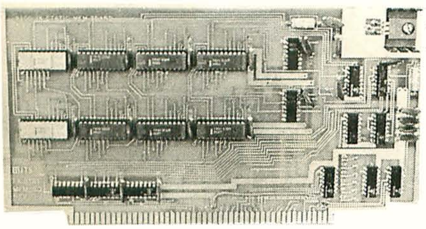
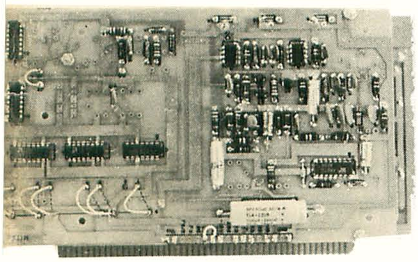
ISSUE # 2 OCTOBER 1975

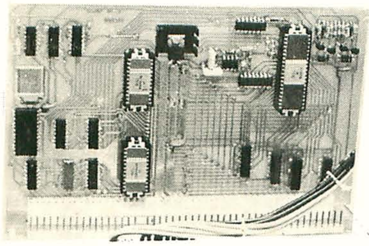
BYTE

$1.50

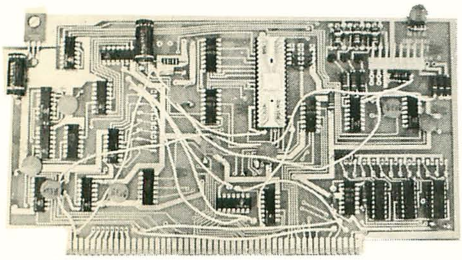
the small systems journal

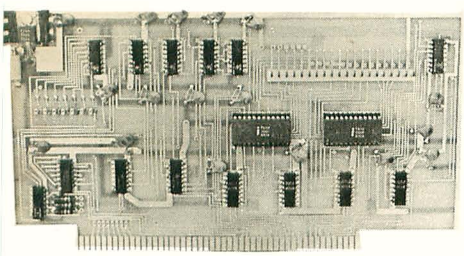


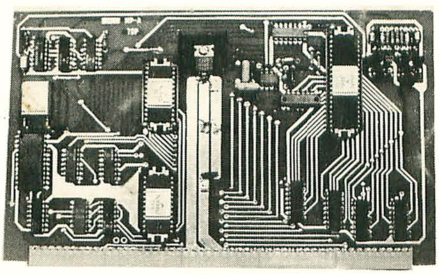


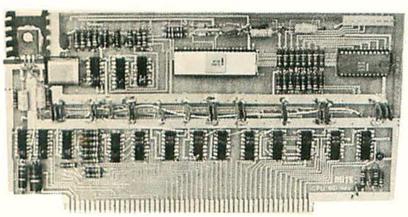


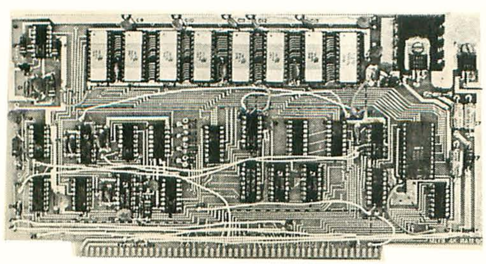
* Build a GRAPHICS DISPLAY
* TELEVISION INTERFACE
* NOVAL ASSEMBLER
* KLUGE HARP: Music?
* LIFE Line: Game





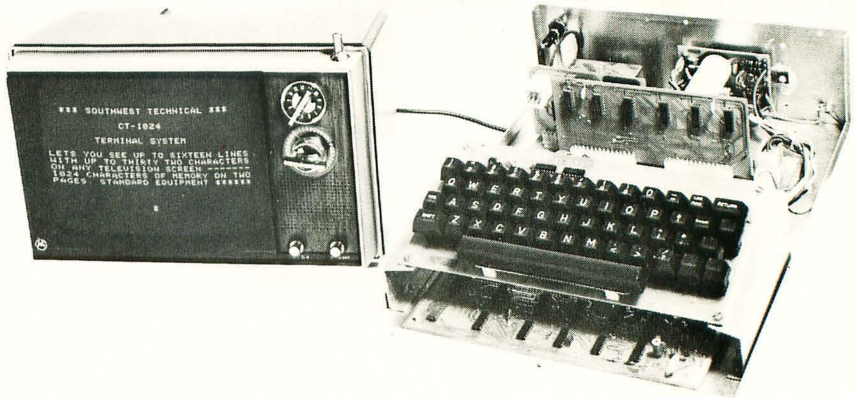






Computers - The World's Greatest Toy!

SWTPC



CT-1024 TERMINAL SYSTEM

When we designed the CT-1024 we knew that there were many applications for an inexpensive TV display terminal system. Even so, we have been surprised at the many additional uses that have been suggested by our customer in the last four months since we introduced this kit.

The basic kit, consisting of the character generator, sync and timing circuits, cursor and 1024 byte memory gives you everything you need to put a sixteen line message on the screen of any TV monitor, or standard set with a video input jack added to it. Input information to the CT-1024 may be any ASCII coded source having TTL logic levels. Two pages of memory for a total of up to one thousand and twenty four characters may be stored at a time. The CT-1024 automatically switches from page one to page two and back when you reach the bottom of the screen. A manual page selector switch is also provided. The main board is 9½ x 12 inches. It has space provided to allow up to four accessory circuits to be plugged in. If you want a display for advertising, a teaching aid, or a communication system then our basic kit and a suitable power supply is all you will need.

CT-1 TERMINAL SYSTEM with MEMORY KIT........................$175.00 ppd

Power supply kit to provide + 5 Volts @ 2.0 Amps and - 5 Volts, -12 Volts @ 100 Ma. required by the CT-1 basic display system.

CT-P POWER SUPPLY KIT........$15.50 ppd

A very nice convenience feature at a very reasonable cost is our manual cursor control plug-in circuit. The basic kit allows you to erase a frame and to bring the cursor to the upper left corner (home up). By adding this plug-in, you can get Up, Down, Left, Right, Erase to End of Line and Erase to End of Frame functions. These may be operated by pushbutton switches, or uncommitted keyswitches on your keyboard. Although not essential to terminal operation, these features can be very helpful in some applications.

CT-M MANUAL CURSOR CONTROL KIT..$11.50 ppd

If you plan to use your terminal with a telephone line modem, or any other system that requries a serial data output; you will need our serial interface (UART) plug-in circuit. This circuit converts the ASCII code from a parallel to a serial form and adds "Start" and "Stop" bits to each character. The standard transmission rate for this circuit is 110 Baud, but optional rates of 150, 300, 600 and 1200 Baud may be obtained by adding additional parts to the board. The output of this circuit is an RS-232 type interface and may be used to drive any type modem, or coupler system using this standard interface.

CT-S SERIAL INTERFACE (UART) KIT...$39.95 ppd

If you are using the CT-1024 as an IO (input - output) device on your own computer system, you will probably want to connect it to the computer with a parallel interface system. A direct parallel interface allows for much faster data transmission and reception and is basically a simpler device than a serial interface system. Our parallel interface circuit contains the necessary tristate buffers to drive either a separate transmitt and receive bus system, or a bidirectional data bus system. TTL logic levels are standard on this interface. Switch selection of either full, or half duplex operation is provided. The terminal may write directly to the screen, or the computer may "echo" the message and write to the screen.

CT-L PARALLEL INTERFACE KIT..$22.95 ppd

We would be happy to send you a complete data package describing the CT-1024 and a achematic. If you want this additional information, circle our number shown below on your reader information service card. The CT-1024 kit has complete assembly instructions with parts location diagrams and step-by-step wiring instructions. If you would like to check the instruction manual before you purchase the kit, please return the coupon with $1.00 and we will rush you the manual and the additional data mentioned above.

MAIL THIS COUPON TODAY

☐ Enclosed is $__________ ☐ or Master Charge # __________

☐ or BankAmericard # __________ Card Expiration Date __________

☐ CT-1024 Kit ☐ CT-M Cursor Control Kit

☐ CT-S Serial Interface Kit ☐ CT-L Parallel Interface Kit

NAME __________

ADDRESS __________

CITY __________ STATE __________ ZIP __________

☐ $1.00 Enclosed send manual and data package

Southwest Technical Products Corp., Box 32040, San Antonio, Texas 78284

In the Queue

BYTE #2

OCTOBER 1975

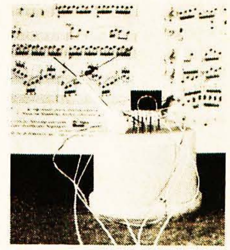
Foreground

Background

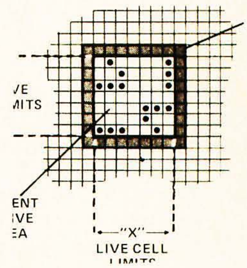
Nucleus

BYTE magazine is published monthly by Green Publishing, Inc., Peterborough, New Hampshire 03458. Subscription rates are $12 for one year worldwide. Two years, $22. Three years, $30. Second class postage application pending at Peterborough, New Hampshire 03458 and at additional mailing offices. Phone: 603-924-3873.

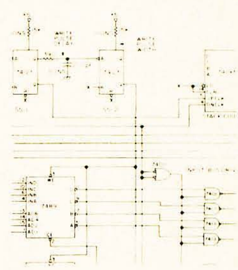
Address editorial correspondence to Editor, BYTE, Box 378, Belmont MA 02178.



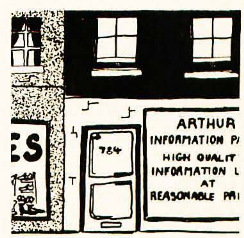
p. 14



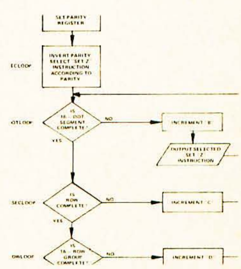
p. 34



p. 52



p. 58



p. 70

The Sphere System...

Tomorrow's Computer Today!!!

IT WILL give you a competitive edge in many new micro-computer professions.
IT WILL do your payroll and inventory for less than ANY competitor.
IT WILL speed education in computer science using stand alone or disk basic.
IT WILL monitor and protect your home or business when you can't.
IT WILL maintain your mailing list, balance your checkbook, or edit your resumé.
IT WILL present new challenges long after your kit assembly project has ended.
IT WILL allow you to expand to the largest set of peripherals and software available in today's low cost computer market.

AND . . .
IT WON'T BREAK YOUR BANK !!!

MORE →

Are you?
- □ *an educator*
- □ *a hobbyist*
- □ *an OEM . . . QTY/YR* ____
- □ *looking for product to sell.*

What is your application

____

____

____

____

send info to ____

____

____

ZIP ____

Fold Back Here

STAMP HERE

SPHERE CORP.
791 South 500 West
Bountiful, Utah 84010

cut here *cut here*

Clip out and mail coupon or call (801) 292-8466

			Before Sept. 30	After Sept. 30
		KITS		
SYS 1	*4K*	*Byte computer with TV terminal, keyboard, and ROM monitor*	*$ 650*	*$ 860*
SYS 2	*4K*	*Byte computer with TV terminal, keyboard, ROM monitor, and audio cassette interface.*	*750*	*999*
SYS 3	*20K*	*Byte computer with TV terminal, keyboard, ROM monitor, audio cassette interface, and extended basic programming language.*	*1345*	*1765*
		NEW FEATURES		
PEN		*Light pen kit for CRT board*	*80*	*100*
BAS		*1K micro-basic prom set (a modified basic language that makes your computer talk basic without loading from cassette)*	*90*	*140*
RAC 2		*Attractive plastic chassis with CRT display and acoustic coupler.*	*270*	*300*

SPHERE

791 South 500 West, Bountiful, Utah 84010 (801) 292-8466

Carl Helmers

Editorial: Functional Specifications "The Home Brew Voder"

A whole line of home computer experimentation can be started by the person who designs – and writes up as an article for BYTE – a simple and practical circuit to generate speech output from phoneme codes in a program. The output problem in digital speech representation techniques is well within the range of a microcomputer system and inexpensive digital to analog conversion methods (such as the Motorola 6-bit el cheapo MC1406 DACs).

Consider, for instance, a brute force technique in which the voice info is encoded as 3-bit (eight level) quantities which are sent out at a rate of once every millisecond for conversion. (Rate: 3000 bits per second.) A typical voice "phoneme string" in the micro's memory would specify data for maybe 100 milliseconds. This would require a total of 100 data values or 300 bits, stored in a packed bit string format in 38 bytes plus one overhead byte for string length. If one considers a reasonable vocabulary of speech elements, say 100 basic sounds, the data requirement is thus 3900 bytes – well within the memory budgets of many BYTE readers' systems. The thruput required to output the voice bytes is also well within a micro's capacity – 1000 microseconds is a lot of time to fool around with. It's even long enough for a tortoise of a computer like the 8008 to do enough bit diddling to prepare a command code for a 3-bit DAC port.

Now, what's all the purpose to the home voder? Well, if you can't think of a use for it, I'll supply a couple of suggestions. Wouldn't it be neat to put up a home security system using your micro in which the burglar gets scared out of his wits by a threat issued in a computereeze voice while lights are flashing and other ominous things are happening? Or in the area of gamesmanship – when the game program gets erroneous input, have your program issue the text of the message "foul!" Or, getting less exotic, simply make yourself a calculator which will literally read out the answers to your problems as well as show them on an output screen. Getting exotic again, suppose you make a Star Trek oriented space war game. You really should have the "ship's computer" give status reports aurally to make the game more exciting – perhaps coupling in a few bells and whistles (literally) in the form of special purpose synthesizers of photon torpedo, phaser, transporter and other sound effects Trekkies know and love.

Then there is the ultimate application – making a higher grade voice synthesizer which can sing, literally, so your machine can play vocal music.

How can such audio peripherals be brought into existence? One way is through the encouragement I can give by publishing articles on various approaches – the design articles of readers which make BYTE an essential publication for the home computer experimenter. A second way is for the various entrepreneurial readers in the audience to take a cue from this little essay and get working on packaged products in the audio output line which the majority of readers will be able to put together and program.

The essential elements of the inexpensive voice synthesizer are the familiar hardware and software combination: The hardware part is the n-bit (3 will do) DAC output port (and latch) which converts internal codes into one of 8 voltage levels. The DAC in turn will drive an active low pass filter (op amp) and a power amplifier for the speaker. If you get fancy, a second DAC could be used as a gain control output (with an appropriate 8-bit latch for storage) – and the 5 unused bits of the typical speech value word could be used to control 5 additional bells and whistles. The software of the application is in two segments. There is the machine-independent data table which specifies the basic phoneme information – and the rules for combination into understandable words. Then there is the machine-dependent programmed "talker" routine which accepts an "n-byte" character string with phoneme codes and synthesizes the phonemes one by one from the data table by outputting the selected series of 3-bit codes in real time. In a design article, "talker" would be specified functionally in a high level language along with a global flow chart, and the phoneme data information would be specified as a table in hex and/or octal codes. So here you have the idea – let's see what BYTE readers can come up with in the way of articles on the subject.

(Oh yes, one parenthetical note – the proof of the putting is in the speaking. Prospective voder designers should send along a tape recording of their design in operation – preferably speaking the first two lines of Lewis Carroll's poem "Jabberwocky," the universal test string:)

A Need in Search of a Product

Have you ever run a business, newsletter or club which requires periodic mailing of information? Have you ever tried to maintain a mailing list? The purpose of this short essay is to identify an opportunity which exists for the entrepreneurial persons in BYTE's readership – the opportunity to create and market a specialized mailing information system using contemporary technology in the form of microcomputers and inexpensive peripherals. Here is what you have to compete against in the general market – the free market of all the possible solutions to the problems of mailing labels.

Multipart Carbon Forms

My old standby in the mailing list area is the multipart carbon form which is manufactured by Dennison among others. These forms hold 33 names per sheet and come with four parts to reduce typing and retyping of names. They come with water base glue backing and are perforated – but the main problem is typing. The 33 name per sheet figure only holds if you don't make any typing errors! The degree of automation of this system is usually zero, unless you have a power typewriter and are using the carbon forms to save output. Cost is also low – paid for in typing time of course – at about $2 to $3 for a package of ten. No competition – labor intensive.

Spirit Duplicator Methods

The next step up in the world of office automation is a system involving spirit duplication stencils – the blue ink smeary reproduction that was in vogue for high schools and grade schools before Xerox and its imitators became so widespread. A Sears version of this system, per their office products catalog, costs from about $70 upwards. Similar units run up to the range of about $200 to $300 and address master blanks for the "Elliott" system cost about a dime each. This type of system has a higher degree of automation (retyping of names is much less frequent) – but still involves a costly "use-only-once" part, the stencil.

Addressograph-Multigraph

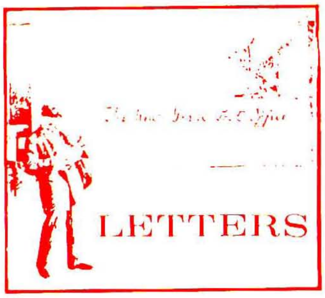
The next step up in cost – a much more permanent system for large usage – is the Addressograph-Multigraph type of system which uses metal plates prepared in advance. These systems have a much longer lasting and more expensive label master blank – and minimal systems can be had in the $800 range and up. The system involved is essentially the same as the duplicator style systems – but more costly, due to fancier equipment, some automation of envelope handling and longer lasting media. There is still no automation of the typing and related information handling. Here is where the new computer systems will begin to effectively compete.

The Hypothetical Small Computer Addressing System

A "trivial" application of the microprocessor technology which can be assembled by any of BYTE's more experienced readers is a hardware/software system consisting of the following items:

1. Microprocessor (8-bit) with 1024 bytes RAM, 2048 to 4096 bytes ROM program, three serial ports, one 8-bit parallel output and one 8-bit parallel input.
2. Adding machine tape ASCII printer – accepts parallel output and prints it as characters on rolls of tape.
3. ASCII input keyboard – parallel input of text and commands.
4. Triple audio cassette interface capable of two input operations and one output operation simultaneously.
5. Edit/Merge program – software for editing of "current label" records held in RAM, using the keyboard for commands and text input, using the printer for output. Allow optional input from one tape port, output to a second tape port, with automatic sequence checking to maintain a sorted sequence on the files.
6. Update/Merge program – software for "batch" changes to the file, in which a previously prepared (and sequenced) update tape is merged with an old tape to create a new output tape.
7. Print program – software to print the file – either as an unconditional dump of all labels, or a selective dump such as "every nth name" or "names with zips 07932 to 07860."
8. A Percy Wing Machine. This is a hand-operated gizmo which costs about $100 and is used to automatically apply labels from the roll of printout – slicing each label off the roll and automatically applying it with glue to the mailing piece.

Put together a packaged product based on these ideas, and it could most likely be sold in the $1500 to $2000 range, supplying a nice profit margin and a product which competes effectively with the Addressograph-Multigraph "systems," yet provides automatic features and a much more compact storage method (cassette tape) for lists of moderate size.



"IT'S ALL IN THE EPHEMERIS . . ."

Dear Mr. Helmers,

I have sent in a subscription order for BYTE. It seems to be what I'm looking for. Already I have been looking at surplus ads for CPUs with an eye to making my own digital computer though "simple minded" it would probably have to be.

Among other things, I am a ham and a consulting engineer for EBASCO. What I have in mind is to have available a programmable computer to solve filter design problems (and antenna/feedline impedance problems parametrically). *However* – as a first question: Do you think it may be in the realm of a "non-computer expert" to make a home built computer solve weather satellite orbits and read out azimuth and elevation information for tracking purposes? I can feed in very accurate time information. I don't know yet how to get from "here" to "there."

W. J. Byron
New York NY

You have an interesting question . . . For the first portion, how to solve the analytical equations of engineering, there is a range of solutions depending upon your purposes, budget and other factors. The simplest approach, with the least amount of money and the least amount of computer technology "learning" experience is to simply go out and buy one of the new HP-25 calculators. You'll find it quite capable of solutions to a large number of numerical analysis problems for engineering – although it's adequate for extensive linear algebra and matrix data calculation. My associate Chris Bancroft has been using the slightly more powerful HP-65 for more than a year now to arrive at exact analytical (and very predictive) solutions to engineering problems in applied electronics – and the HP-25 should be able to perform similarly on many problems. But a pre-packaged calculator may be "no fun" – and certainly is not useful for anything other than calculation. Further, the programming of a really complicated satellite position algorithm (e.g., adding in second and third order perturbations of the earth's gravitational potential) may be beyond the range of a simple hand calculator without multiple mental overlays.

The problem of simple celestial mechanics computation is well within the range of a small home microcomputer system which contains perhaps 4k bytes (or equivalent) and the facility to do overlay programming using at least one audio cassette drive. In order to do this kind of calculation, however, I am making the following assumption: you already have the analytical solution to the problem in the form of an algorithm specification. (In case you don't, maybe a reader with a bent for applied celestial mechanics, coordinate transformation problems and numerical integration might be so kind as to supply a solution to this problem either to you privately or for publication.)

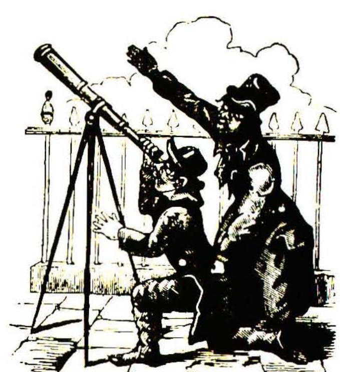
In order to program the typical three-space navigation problem, assuming well determined static orbital elements, the following items are required:

1. A computer with software for SIN, COS, TAN and ARCTAN trig functions (you can get this set from an interface calculator chip).

2. Vector-matrix algebra subroutines for three-space. Since you are talking about static orbits (no active control altering parameters), you don't need much in the nature of the more advanced error correcting algorithms.

3. A floating point arithmetic package to go along with the computational requirements of #1 and #2. This should probably include data conversions to integer and character string forms for convenience of programming.

4. An orbital model which uses a given set of ephemeris information at some time "t" and calculates new satellite position at time "t + Δt" later. (The "now" position.)



5. An output model which uses various coordinate transformations to turn the "now" position in "geocentric coordinates" into an apparent position on the unit sphere for a known latitude, longitude and time. This is the right ascension and declination of an optical telescope or equivalent terms for your radio antenna.

The place to look for detailed information is an advanced undergraduate or graduate level book on classical mechanics, spacecraft navigation or astronomy if you want to find info on the calculations. One possible source of information, mentioned to me by Bob Baker of Littleton, Mass., is a radio amateur organization called the Amateur Satellite Corporation – AMSAT, Box 27, Washington DC 20044. Bob tells me that their newsletter frequently mentions computational algorithms for the OSCAR amateur radio satellite – algorithms which could be adapted to any satellite orbital elements. Another possible source of information is the HP-65 Users Club run by Richard Nelson – in scanning through his back issues recently I saw several listings of names of programs sounding suspiciously like what you want, e.g., "Orbital Element Determination" and the like. (HP-65 Users Club, 2541 West Camden Pl., Santa Ana CA 92704.) While by no means a complete design, I hope this information proves useful to you.

. . . CARL

EVOLUTION NOT EROSION

Dear Mr. Helmers,

I am a subscriber to ECS magazine and was a little uneasy to receive the announcement of its transformation into "BYTE". I hope this is an evolutionary development, and not an erosion into the murkiness of total commercialism. I have very much enjoyed and benefited from ECS.

Thanks for the help, and good luck in your new venture.

Duane L. Gustavus
Denton TX

I think I can sympathize with your uneasiness about ECS Magazine's transformation into BYTE. Yes, it is indeed an evolutionary transition, in several respects. First, if you look in the first issue you will find a much larger and more varied editorial content than I was ever able to achieve in the course of preparing my self-published 24-page photo offset magazine. Second, by providing a place for commerce – a free market – it serves as a unifying element in the whole sphere of the computer hobby endeavor. Throughout history, it is the market place which has sustained the progress in ideas, and technology which has brought the human race from crude wheels to $20 bus-oriented microcomputers. Third, there is the element of professionalism in format and execution. BYTE is being put together by a fine organization of craftspeople who take pride in the work which results. Sure, the magazine is commercial and has advertisements – but then, don't you work for yourself, some employer, or other agency? This pride in the quality and value of work extends throughout the publishing operation I have joined – and it will be the element which makes the reputation for the magazine as a source of ideas and fun.

. . . CARL

"BILL ME"

Dear Sir:

Yes, I'll byte, but I think I've been taken. I was a subscriber of ECS and was told that subscription was okay for the charter BYTE publication; I guess that promise ... and ten bucks can also buy a year's subscription. I've been bit (for more than the ten bucks) by others feeding off the crumbs of micro machine data, so can't begrudge you yours.

I am eternally hopeful for BYTE, but not optimistic. I have been a ham for a long time (23 years), and Mr. Green is not my favorite; also I didn't think Mr. Helmer's ECS info was very good. I agree, however, that there "sure is a need for a good magazine," so hope BYTE can serve that need. Let's hope it's better than 73.

Please bill me for the lousy sawbuck.

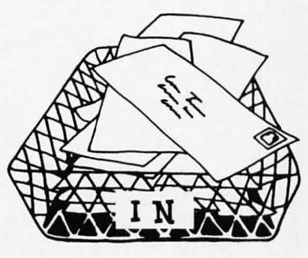
C. Southard WAØIOT
Cedar Rapids IA

I am printing your letter in BYTE for a reason, a matter of principle if you will: I am personally responsible for the editorial content of BYTE magazine, and have my reputation on the line as a result. I believe that after you have read your first issue of BYTE you will find it well worth a $10 which you had no reason to send in at this time other than (perhaps) a negative attitude and a promotional circular which came your way as a result of one of our mailing lists. I have a number of items of unfinished business in connection with the M. P. Publishing Co. operation I was running in my spare time until BYTE started – one of the first such items was a condition of the arrangement I made with Green Publishing that all ECS subscribers should be picked up by BYTE on a two issues for one basis. Thus since your ECS subscription of $21 was fulfilled by mailings of 10 issues, the remaining two ECS issues become four BYTE issues. Adding to that your inadvertent resubscription gives a total of 16 BYTES . . .

OK, you say, "he promises 16 BYTES – will I get them?" You can best evaluate that after you've gotten BYTE #16. The aim of BYTE magazine is – as is the aim of the large corporation for which you work – to turn a reasonable profit in the long run and provide enduring and satisfying work for those connected with the enterprise. It can only do so by providing good service to its customers – the readers and advertisers who patronize the magazine. BYTE is very much a market phenomenon, and cannot exist unless it maintains a readership of intelligent and active persons such as yourself. If BYTE were to consistently turn you off – as well as others on a large scale – we'd be heading into bankruptcy faster than a Penn Central express train should be running.

I invite you or any reader who thinks he or she is getting a bad deal on the magazine to write me personally at any time. I won't promise to publish all such letters – yours, Mr. Southard, is published by distinction of being the first – but I will endeavor to answer each one personally.

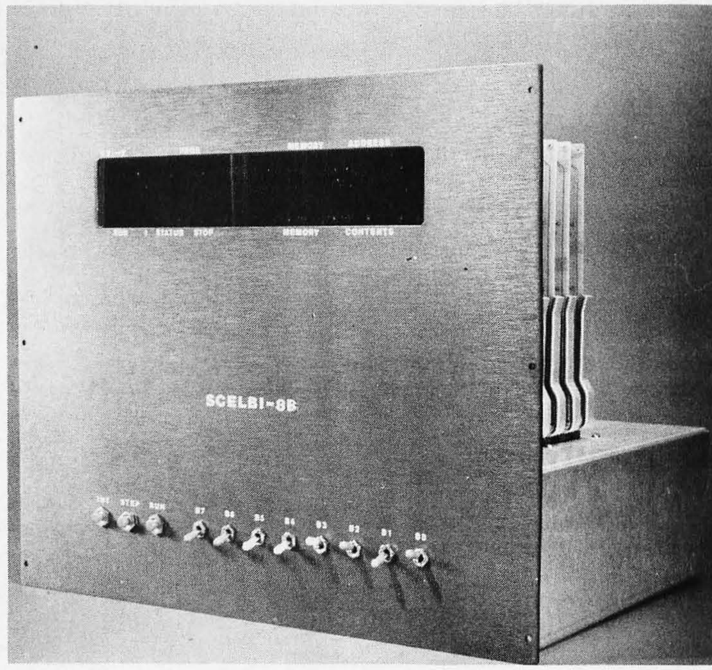
As to technical content, you can peruse the first issue, then make a judgment. I will endeavor to produce the best possible magazine by selecting the best possible articles. I am not about to ignore the biggest multiprocessor system of them all – the human race ... all the people in the readership of BYTE who will be coming up with ideas for articles and submitting them will set the level of much of the material available for publication. If you don't like my personal work, kindly give me the courtesy of stating why – I know that the ECS Series articles I previously put out had many flaws. I have heard some good and some bad evaluations of the ECS *Series and* ECS Magazine *items.*



I will not however disown anything I have done – and that series of self-published articles is my product with all its flaws and imperfections. If you wish to exercise the moneyback guarantee, let me know . . . and send back your copies to fulfill your part of the exchange.

. . . CARL

WHAT SINGLE ELECTRONIC MACHINE CAN BE USED TO PERFORM/CONTROL ALL THE FOLLOWING TYPES OF SERVICES?



- Send morse code
- Control repeater stations
- Operate as a calculator
- Receive/send/buffer data between a wide variety of communication devices
- Monitor instruments
- Control machines
- Sort/compile data
- Test other devices
- Play games

the SCELBI-8B MINI-COMPUTER CAN!

SCELBI COMPUTER CONSULTING, INC. - The company that pioneered in producing the small computer for the individual user with the popular SCELBI–8H, now brings you the new SCELBI–8B with increased capability!

Like the former SCELBI–8H, the SCELBI–8B is built around the amazing '8 0 0 8' "CPU-on-a-Chip" which has been revolutionizing the electronics world.

However, the NEW SCELBI–8B offers extended memory capability at reduced cost! It is directly expandable to 16,384 words of RAM/ROM/PROM memory. This increased memory capability now means the user has the potential in a small and compact computer to support compiler type languages, manipulate sizable data bases for business and scientific applications, and support a wide variety of programs including those that take advantage of external mass memory storage devices.

The NEW SCELBI–8B still retains the outstanding features of its predecessor. Decoding logic for 8 Output and 6 Input Ports is built into the basic computer. Plug-in capability for I/O devices is provided on the chassis. A unique, simple to operate console that utilizes just 11 switches on the front panel makes the SCELBI–8B a pleasure to use.

The NEW SCELBI–8B is backed by a line of low cost SCELBI interfaces which currently include: an interface that turns an oscilloscope into an alphanumeric display system, low cost keyboard and TTY interfaces, and an interface that turns a low cost audio tape cassette into a "Mag–Tape" storage and retrieval unit.

Last, but certainly not least, SCELBI has a wide selection of software ready to run on the NEW SCELBI–8B including: Editors, Assemblers, calculating programs, I/O and general utility routines. Additionally, SCELBI produces publications that can show you how to develop your own custom tailored programs.

The NEW SCELBI–8B is **available** NOW. (We have been delivering since June!) It is available in three forms. Ultra-low cost "Unpopulated" card sets with chassis kits starting at $259.00*. Complete parts kits for a 1,024 word mini-computer as low as $499.00*. An assembled and tested 4,096 word computer is just $849.00*. Interfaces, accessories, and software sold separately.

(*Domestic prices.)

(Prices, specifications and availability subject to change without notice)

Literature available for S.A.S.E.

SCELBI COMPUTER CONSULTING INC.

1322 REAR BOSTON POST ROAD
MILFORD, CONNECTICUT 06460

Clubs Newsletters

LA Update

Thanks to the efforts of Derek McColl, I've received a further update on the Los Angeles club activities . . . a very active bunch of people. Derek sent me a copy of the Southern California Computer Society *Interface* – Volume 1, #0 (do I detect an algolmaniac at work?) for August. The motto in the heading line reads "an announcement for computer hobbyists designed to connect people and ideas . . ." – and it lives up to that billing in its six pages of typewritten copy.

The person who is handling the administrative details of SCCS for the time being is Hal Lashlee, who can be reached by phone at 1-213-682-3108, or by mail at PO Box 987, South Pasadena CA.

The following topics and concerns were drawn from the suggestion boxes of a brief survey form handed out to members at the first meeting of SCCS (and printed in Volume 1 #0 of *Interface*):

– Members would like to see a computerized clearinghouse for computer hobby information.

– One purpose of the organization should be mutual assistance with specific problems.

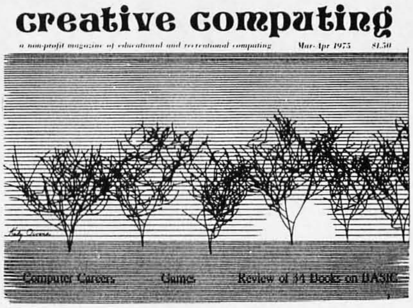
– Hardware procurement by group purchases.

– Standardization.

– The club might act as a brain pool for small business needs.

– The usual club type activities of social meetings, lectures, seminars, workshops, public service, etc.





Creative Computing is the name of a magazine which is edited and published by David H. Ahl. The motto of the operation is "a non-profit magazine of educational and recreational computing." *Creative Computing* is published bi-monthly, printed in a saddle-stitched 8½ by 11 format (similar to BYTE) with 60 pages in a typical issue. The editorial content is heavily oriented toward information useful in an educational context. David Ahl was formerly the Educational Marketing Manager at Digital Equipment Corp., where he was responsible for the creation of DEC's *EDU* publication. To quote from his editorial in the March-April 1975 issue of *Creative Computing:*

"Over the years *EDU* flourished and grew into a 48-plus page magazine. However there were certain aspects of educational computing which *EDU* could not satisfactorily address. In particular, school users, both college and elementary/secondary, need more classroom activities, exercises, problems and ideas than are available in textbooks and other magazines. Also there ought to be a discussion of the social aspect of the computer, its effect on jobs, medical care, privacy and the like. Furthermore, what about the user of non-DEC computers? Clearly to be responsive to these needs another vehicle was needed. Thus *Creative Computing* was born . . ."

The various issues I have seen to date include numerous puzzles, BASIC games, and articles on computers, computer education and computer careers. To order a subscription, send $8 for one year or $21 for three years to *Creative Computing,* Box 789-M, Morristown NJ 07960. . . . CARL

Notes from the Garden State via the Goethels Bridge

ACSNJ's second meeting was held at the Union County Technical Institute on July 18, 1975. The meeting was presided over by Bohdan in the absence of Sol Libes. Thirty hobbyists showed up including 8 new members. In deference to Stephen Gray, founder of the original ACS, we are considering changing our name to the New Jersey Amateur Computer Group (NJACG).

Roger Amidon gave a presentation on the UART. Marty Nichols held a discussion on the differences between 8008 and 8080 microprocessors. Andy Vics talked about his experiences with the construction and operation of TVT 1 and TVT 2. Wayne Ahlers showed us his octal keyboard built around PEs 12-74 low cost computer terminal. Literature and other information was disseminated before and after the formal meeting.

Later that night, a small group visited Roger Amidon's site. In addition to his 16k Altair, TTY and magnetic and paper tape peripherals, Roger K2SMN also has a home brew RTTY controller which is affectionately named Spider. (If I can ever get a picture of it, you'll understand why.)

The third meeting of the NJACG will be held in September, not August, on the third Friday, September 19, 1975, at the Middlesex County College. For more information contact George Fischer, 1-212-351-1751.

Oklahoma City Club

In Oklahoma City OK, Bill Cowden reports that he is organizing a computer enthusiasts' club. Contact Bill at his home address, 2412 SW 45th, Oklahoma City OK 73119.

Add a Kluge Harp to Your Computer

by
Carl Helmers
Editor, BYTE

One of the most interesting computer applications is that of electronic music. This is the use of software/hardware systems to produce sequences of notes heard in a loud speaker or recorded on magnetic tape. The idea of generating music – if well done – is of necessity complex. If I want to put my favorite Mozart piano sonata into an electronic form, I'd have to record a very large number of bits in order to completely specify the piece with all the artistic effects of expression, dynamics, etc . . . The magnitude of the problem can be intimidating. But, never let a hard problem get in the way of fun! Simplify the music problem to one channel of melody, and you can use a virtually bare CPU with a very simple peripheral to play music.* The combination of the CPU with this simple peripheral is what I call the "Kluge Harp" – a quick and dirty electronic music kluge.

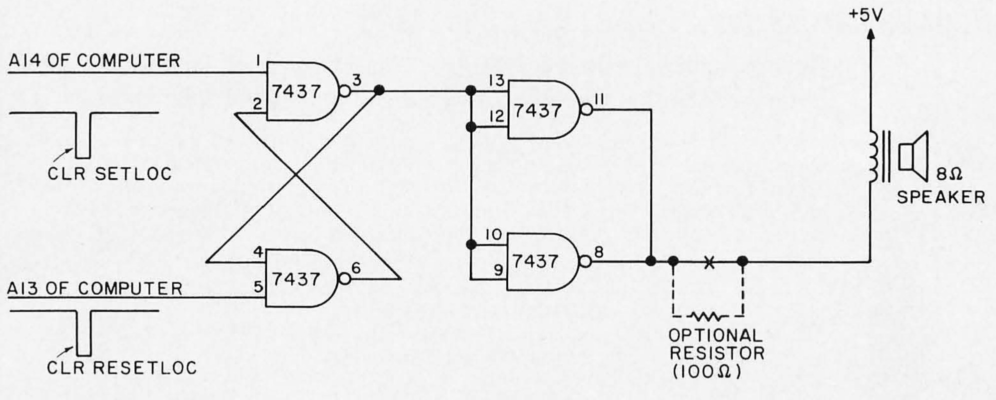
I invented this electronic music kluge to answer a specific problem: I had just gotten a new Motorola 6800 system's CPU, memory and control panel up and running. The next problem (since I wasn't using the Motorola ROM software) was to make a test program which could be loaded by hand. By combining a little imagination, my predilections for computer music systems and an evening getting the whole mess straightened out, the Kluge Harp resulted. While the program and schematic are specific to the system I was using, the *idea* can be applied to your own system just as well.

(*ALTAIR owners: Write an 8080 version of this program and your machine can do more than blink its lights.)

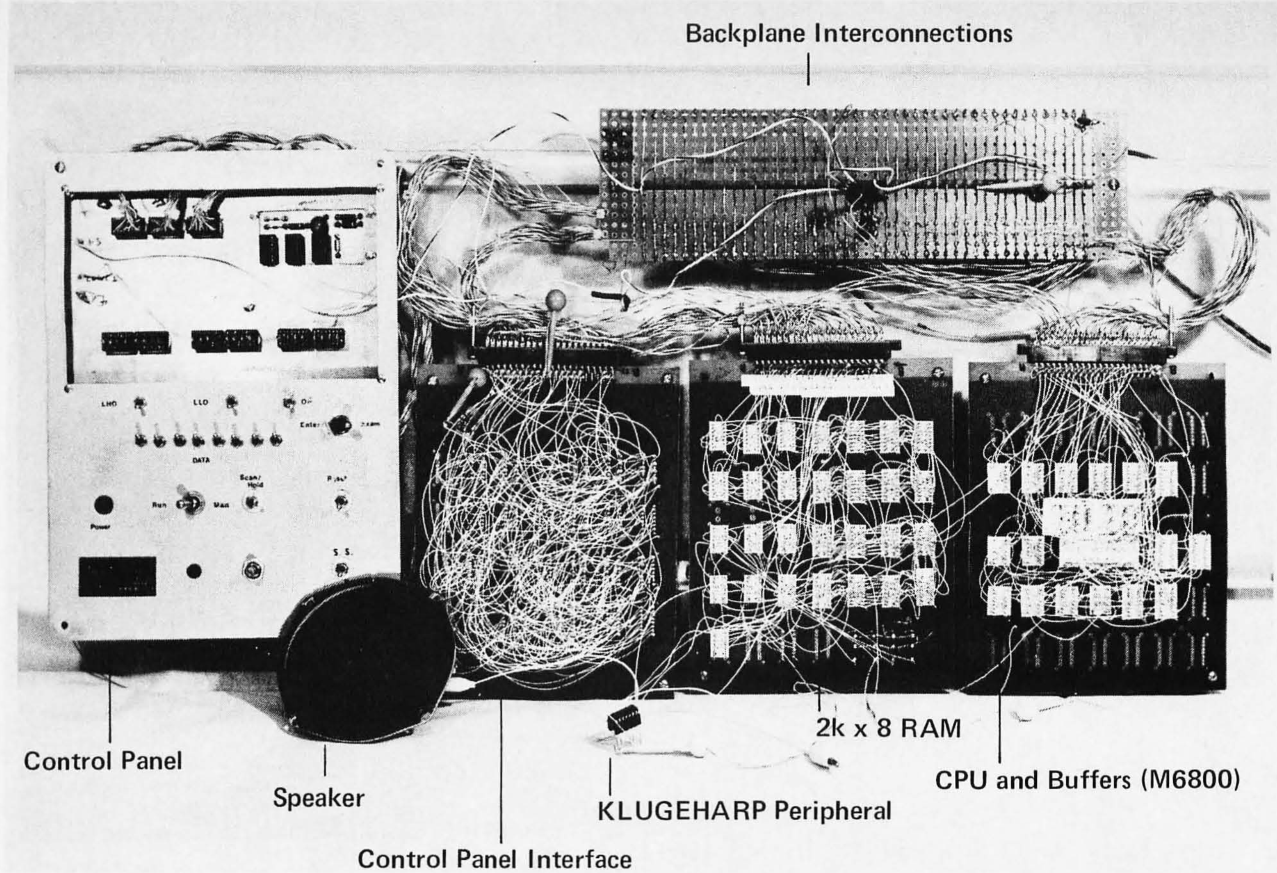
The Kluge Harp Hardware

The hardware of a Kluge Harp is simplicity at its essence. The peripheral is driven off two "un-used" high order address lines (I used A14 and A13), and consists of a set-reset flip flop. A program running in the computer alternately will set and reset the flip flop by referencing one or the other of two addresses. These addresses are chosen so that the address lines in question will change state, actuating the set or reset side of the flip flop. A "note" at some pitch consists of a delay loop in the program followed by instructions to change the state of the flip flop. Since the same count is used for the two halves of a complete cycle of the note, this will produce a perfect square wave. The actual music program organization is a bit

Fig. 1. The Kluge Harp Circuit . . . minus computer.



Michael Gipe



The Kluge Harp peripheral and the KLUGEHARP program were concocted in order to test out a Motorola 6800 system's operation. This photo shows a test bench mounting of the three main cards and control panel. The Kluge Harp peripheral, such as it is, is the single isolated wire wrap socket in the foreground, with wires dangling from connections on the CPU card.

more complex and is described in detail below.

Fig. 1 illustrates the hardware as implemented in my system. The 7437 circuit is used to form the NAND gate flip flop. This flip flop in turn drives a parallel combination of the two remaining 7437 gates, acting as a buffer. The output of this buffer is used to drive the speaker; an 8 Ohm 5" speaker produced more than adequate volume. (A 100 Ohm resistor in series will limit the volume level to spare the ear drums.)

Generating Music With Program Loops

Fig. 2 illustrates the basic concept of the one-channel music generator, expressed in a procedure-oriented language for compactness. The main program loop begins at line 2 of the listing – "DO FOREVER" means repeat over and over again all the lines of code down through the "END" at the same margin, found at line 17. This is the main loop used to cycle through the SCORE stored at some point in memory as pairs of note selection/length data bytes.

Lines 3 to 4 compute the "next" pointer to the SCORE – incrementing NOTER by 2. Then LNGTH is set equal to the second byte of the current pair, SCORE (NOTER+1). The length codes are taken from Table I along with note codes when you set up a SCORE, and represent a fixed interval of time for the note in question, measured as the number of cycles.

Line 6 begins a note length loop which extends to line 14. This "note length" loop repeats the generation of the note a number of times indicated by the length code just retrieved. The note generation is accomplished by delaying a number of time units (CPU states) set by the pitch code found at SCORE(NOTER), then changing the state of the output flip flop and repeating the process. The loop at lines 8-10 counts down the pitch code and has a fixed delay multiplied by the pitch code to give the time for one half cycle of the desired frequency. Lines 11 to 15 change the state of the Kluge Harp output device (0 to 1, 1 to 0) – remembering in the software location IT what the previous state was.

Generating Codes

Table I is a reference table of 21 notes "roughly" spaced at equal intervals on the well tempered scale. The integer numbers in the "divide ratio" column were determined using the prime number 137 as an arbitrary starting point and calculating the integer closest to the result of the following formula:

$$r_n = e^{(\ln(137) + n \ln(2)/12)}$$

Where e is the usual mathematical number 2.717 . . . and the natural logarithm of x (base e) is indicated by ln(x). This is the standard mathematical calculation of the musical "well tempered" scale – the 8-bit approximation used by the Kluge Harp is not perfect by any means, but comes close enough for the purposes of this project.

The length count columns are determined based upon the assembly language generated code for this

Fig. 2. The KLUGEHARP program specified in a procedure-oriented computer language.

```
KLUGEHARP:   PROGRAM;
    DO FOREVER;
        NOTER = NOTER + 2;
        IF NOTER = NOTEND THEN NOTER = NOTESTART;
        LNGTH = SCORE(NOTER+1); /* SECOND OF TWO BYTES */
        DO FOR I = LNGTH TO 1 BY -1;
            PITCH = SCORE(NOTER); /* FIRST OF TWO BYTES */
            DO FOR J = PITCH TO 1 BY -1;
                /* COUNT DOWN THE PITCH DELAY */
            END;
            IT = IT + (-127); /* SWITCH SIGN BIT OF IT */
            IF IT 0 THEN
                SETLOC = 0; /* SET FLIP FLOP WITH MEMORY REF */
            ELSE
                RESETLOC = 0; /* RESET FLIP FLOP WITH REF */
        END;
    END;
  CLOSE KLUGEHARP;
```

routine, so that for each pitch, the corresponding length count column will measure a nearly identical interval of time. The formula is:

$Lc_n = time / (oh + dt\# pc_n)$

where:

Lc_n = n^{th} length count.

time is the total number of states for one "beat" of the music (e.g., the shortest note).

oh is the overhead of the length counting loop.

dt is the number of states in the pitch count innermost loop.

pc_n is the pitch count for the nth frequency.

Table I shows the divide ratio in decimal, a hexadecimal equivalent note pitch code, and seven

Data assumed by KLUGEHARP:

NOTER: 16-bit (two-byte) address value. Initialize to point to the address of the first byte of SCORE.

SCORE: An array of data in memory containing the code sequence of the music (see Table II). Initialize with the music of your heart's desire or use the example of Table II.

NOTEND: 16-bit address value, the address of the last byte of SCORE (must be an even number).

NOTESTART: 16-bit address value, the address of the first byte of SCORE (must be an even number).

SETLOC: An unimplemented address location which if referenced turns off one bit among the high order address lines, bit 14 in the author's case.

RESETLOC: An unimplemented address location which if referenced turns off one bit among the high order address lines, bit 13 in the author's case.

Data used but not initialized:

LNGTH

PITCH

IT

I, J

Table I. Kluge Harp Synthesizer pitch/length specification codes (HEX).

n	divide ratio	hex note code	Note Length Codes (second byte of pair) 1	2	4	6	8	16	32
-10	77	4D	19	32	64	96	C8	–	–
-9	81	51	18	30	60	90	C0	–	–
-8	86	56	17	2D	5A	87	B4	–	–
-7	91	5B	16	2B	56	81	AC	–	–
-6	97	61	14	29	51	7A	A2	F3	–
-5	102	66	13	27	4D	74	9A	E7	–
-4	108	6C	12	25	49	6E	92	DB	–
-3	115	73	11	23	43	68	8A	CF	–
-2	122	7A	10	21	41	62	82	C3	–
-1	129	81	10	1F	3E	5D	7C	BA	F8
0	137	89	0F	1D	3A	57	74	AE	E8
1	145	91	0E	1C	37	53	6E	A5	DC
2	154	9A	0D	1A	34	4E	68	9C	D0
3	163	A3	0C	19	31	4A	62	93	C4
4	173	AD	0C	18	2F	47	5E	8D	BC
5	183	B7	0B	16	2C	42	58	84	B0
6	194	C2	0B	15	2A	3F	54	7E	A8
7	205	CD	0A	14	28	3C	50	78	A0
8	217	D9	09	13	25	38	4A	6F	94
9	230	E6	09	12	23	35	46	69	8C
10	244	F4	08	11	21	32	42	63	84

Fig. 3. Motorola 6800 Code for KLUGEHARP program.

Address	Data	Label	Opcode	Operand	
F800	FE	KLUGEHARP:	LDX	NOTER	
F801	FA	3:			
F802	00				Add 2 to location in score
F803	08		INX		by incrementing and then
F804	08		INX		saving 16-bit new address
F805	FF		STX	NOTER	
F806	FA				
F807	00				
F808	8C	4:	CPX	#NOTEND	compare against immediate
F809	FC	NOTEND:	(last address of		
F80A	80		SCORE plus 2)		
F80B	26		BNE		Skip if not at end . . .
F80C	03		*+3+2		
F80D	CE		LDX	#NOTESTART	
F80E	FC	NOTESTART:	(first address of		otherwise recycle
F80F	00		score . . .)		
F810	FF		STX	NOTER	save in either case . . .
F811	FA				
F812	00				
F813	FE		LDX	NOTER	This is superfluous!
F814	FA				
F815	00				
F816	E6	5:	LDAB	1,X	
F817	01				
F818	5A	LENGTH:	DECB		
F819	26	6:	BNE		Skip if length remains . . .
F81A	03		*+2+3		
F81B	7E		JMP	KLUGEHARP	Restart piece
F81C	F8				
F81D	00				
F81E	A6	7:	LDAA	0,X	
F81F	00				
F820	4A	FLOOP:	DECA		
F821	26	8:	BNE	FLOOP	
F822	FD		*+2-3		
F823	86	11:	LDAA	#80	
F824	80		(-127)		
F825	BB		ADDA	IT	
F826	FA				
F827	02				
F828	2B	12:	BMI		
F829	05		*+2+5		
F82A	7F	13:	CLR	SETLOC	
F82B	B0		(address with bit 14 off . . .)		
F82C	00				
F82D	20		BRA		
F82E	03		*+2+3		
F82F	7F	15:	CLR	RESETLOC	
F830	D0		(address with bit 13 off . . .)		
F831	00				
F832	B7		STAA	IT	
F833	FA				
F834	02				
F835	7E	16:	JMP	LENGTH	
F836	F8				
F837	18				

Data allocations for KLUGE-HARP:

FA00-FA01 = Current pointer to SCORE, NOTER, which should be initialized to FC00 before starting the program.

FA02 = IT – an arbitrary initialization will do.

FA03-FFF7 = memory area available for SCORE – the example uses FC00 to FC7F and puts the relevant initializations into locations F809-F80A (NOTEND) and F80E-F80F (NOTESTART).

NOTE: In the label column, the numbers followed by colons (e.g., "6:") are used to indicate corresponding places in the high level language version of the program of Fig. 2.

In the system for which this program was written, all active memory is found at addresses F800 to FFFF. Thus for all normal program activity, bits A14 and A13 at the back plane of the system are logical "1". When the location SETLOC (B000) is cleared, the high order address portion changes and bit A14 goes to negative for a short time, setting the Kluge Harp flip flop. When the location A13 is cleared (D000) on an alternate cycle, address bit A13 goes to logical 0 for a short timer resetting the Kluge Harp flip flop . . .

columns of hexadecimal length codes weighted to 1, 2, 4, 6, 8, 16 and 32 unit intervals of time. A note is placed in the score by picking a note code, putting it in an even numbered byte, then placing a length code from the same line of the table in the odd numbered byte which follows it. The actual pitches you'll get from these codes depend upon the details of the algorithm in your own particular computer and the clock rate of the computer. For the 6800 system on which Kluge Harp was first implemented, the lowest note (code F4) is approximately 170 Hz with a 500 kHz clock – and the unit interval of time is approximately 2000 CPU states or about 4 milliseconds.

The hand assembled M6800 code for the KLUGEHARP program is listed in Fig. 3. The mnemonics and notations have been taken from the Motorola *M6800*

Table II. WOLFGANG: Set the content of SCORE in memory to the codes in this table — given for the addresses of the M6800 program version — and KLUGEHARP will play four bars from the classical period.

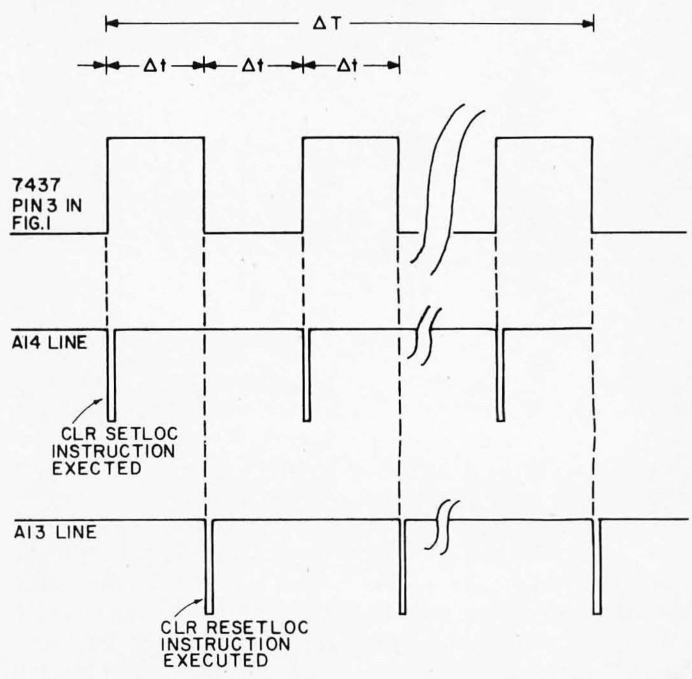
6800 Address	Value		6800 Address	Value	
FC00	9A34		FC40	5B56	
FC02	9A34		FC42	5B56	
FC04	9A34		FC44	5B56	
FC06	9A34		FC46	5B56	
FC08	9A34	Note 1	FC48	5B56	Note 8
FC0A	9A34		FC4A	5B56	
FC0C	9A34		FC4C	5B56	
FC0E	9A34		FC4E	5B56	
FC10	7A41		FC50	664D	
FC12	7A41	Note 2	FC52	664D	Note 9
FC14	7A41		FC54	664D	
FC16	7A41		FC56	664D	
FC18	664D		FC58	4D64	
FC1A	664D	Note 3	FC5A	4D64	Note 10
FC1C	664D		FC5C	4D64	
FC1E	664D		FC5E	4D64	
FC20	A331		FC60	664D	
FC22	A331		FC62	664D	
FC24	A331	Note 4	FC64	664D	Note 11
FC26	A331		FC66	664D	
FC28	A331				
FC2A	A331		FC68	7343	Note 12
			FC6A	664D	Note 13
FC2C	9A34	Note 5	FC6C	7343	Note 14
			FC6E	7A41	Note 15
FC2E	893A	Note 6	FC70	7343	Note 16
FC30	9A34		FC72	7A41	
FC32	9A34		FC74	7A41	
FC34	9A34		FC76	7A41	
FC36	9A34	Note 7	FC78	7A41	Note 17
FC38	9A34		FC7A	7A41	
FC3A	9A34		FC7C	7A41	
FC3C	9A34		FC7E	7A41	
FC3E	9A34		FC80	(end pointer points here)	

NOTE: This program is simpleminded and not at all optimized. As a challenge to readers, figure out a way to make the notation more compact yet preserving the total length of each note.

Microprocessor Programming Manual available from the manufacturer.

While not the greatest musical instrument in the world, the Kluge Harp represents an interesting and challenging diversion. The program presented here is by no means the ultimate in music systems — and can serve as a basis for further experimentation and elaboration. Some challenges for readers: modify the program to change the frequency of the notes without changing the SCORE data; write another (longer) music program which only specifies the pitch code/length information once — and represents the score as a series of one-byte indices into the table of pitch code/length information.

Fig. 4. Timing of the Kluge Harp Output Waveform. Δt is the amount of time spent in the inner loop, and is set by choice of pitch codes. ΔT is the length of the note, measured as a count of half-cycles at its frequency. See Table I for a consistent set of length codes.



ALTAIR 8800 USERS!

Did you know...

- That all our modules are 100% compatible with the Altair 8800 computer, NO modifications necessary!
- That our 4KRA Static Read/Write Memory module doesn't have to lose it's data when you pull the plug!
- That our 3P+S Input/Output module will fully interface two TV Typewriters with keyboards and a modem or teletype at the same time!
- That we make the most powerful alphanumeric Video Display module anywhere!
- That our software is FREE, or close to it!
- That all our modules are truly high quality, computer grade, but that our prices are the lowest in the industry!
- That we have already shipped hundreds of modules on time, and we will continue to deliver what we promise, FAST!

CHECK THE SPECS:

4KRA Static Read/Write Memory

This 4096 word STATIC memory provides faster, more reliable and less expensive operation than any currently available dynamic memory system. The 4KRA permits Altair 8800 operation at absolute top speed continuously. All RAM's (Random Access Memories) used in the 4KRA are 91L02A's by Advanced Micro Devices, the best commercial memory IC on the market today. 91L02A's require typically 1/3 the power of standard 2102 or 8101 type RAM's and each one is manufactured to military specification MIL STD-883 for extremely high reliability. These memories can be operated from a battery backup supply in case of power failure with very low standby power consumption. (Ask for our technical bulletin TB-101 on power down operation.) In short we have done everything we could to make the best 4K memory module in the computer field, and because we buy in large quantity, we can make it for a very reasonable price. Available now.

2KRO Erasable Reprogrammable Read Only Memory Module

With this module the Altair 8800 can use 1702A or 5203 type Erasable Reprogrammable ROM's. The 2KRO accepts up to eight of these IC's for a capacity of 2048 eight bit words. Once programmed this module will hold its data indefinitely whether or not power is on. This feature is extremely useful when developing software. All necessary bus interfacing logic and regulated supplies are provided but NOT the EPROM IC's. Both 1702A and 5203 PROM's are available from other advertisers in this magazine for well under $25. Available now.

3P+S Input/Output Module

Just one 3P+S card will fulfill the Input/Output needs of most 8800 users. There are two 8-bit parallel input and output ports with full handshaking logic. There is also a serial I/O using a UART with both teletype current loop and EIA RS-232 standard interfaces provided. The serial data rate can be set under software control between 35 and 9600 Baud. You can use your old model 19 TTY! This module gives you all the electronics you need to interface most peripheral devices with the Altair 8800, it's really the most useful and versatile I/O we've seen for any computer. Available now.

MB-1 Mother Board

Don't worry any more about wiring hundreds of wires in your Altair to expand the mainframe. Our single piece 1/8-inch thick, rugged mother board can be installed as one single replacement for either three or four 88EC Expander cards, so you don't have to replace your already installed 88EC card if you don't want to. The MB-1 has very heavy power and ground busses and comes with a piece of flat ribbon cable for connection to the front panel board of the 8800. Available now.

VDM-1 Video Display Module

This module is the first real computer terminal display in kit form. Under software control the VDM-1 displays sixteen 64 character lines to any standard video monitor. Characters are produced in a 7x9 dot matrix, with a full 128 character set, upper and lower case plus control characters. Data is accessed by the VDM as a block from any 1K segment within the 65K address range of the 8800 computer. Multiple cursors are completely controlled by software and the display can begin anywhere on the screen (this is great for many video games). When the last line is filled the display scrolls up a line. Powerful editing capabilities are provided with the FREE software package included in every VDM-1 kit. Available in September '75.

SOFTWARE

Our Assembler, Text Editor and System Executive is being shipped now. This software package gives you very powerful Assembly Language capability in the Altair 8800. The Executive and Editor allow you to call programs by name (including BASIC) and then add, delete, change, or list programs by line number. The Assembler provides a formatted symbolic mnemonic listing as well as octal or binary object code from Assembly Language programs written using the Editor. The Assembler also gives valuable error messages to help in debugging those inevitable errors. The Assembler, Editor, Executive Package No. 1 will be available in read only memory along with an expanded Executive and a powerful Interpretive Simulator by October or November of 1975.

We are working on two BASIC Language packages which should be ready by October. One will be a basic BASIC needing about 8K of memory as a minimum and the other will be an Extended version with additional string manipulation, matrix operations and double precision arithmetic capabilities requiring about 12K. Both these packages will be available in Read Only Memory for a reasonable price.

PRICE LIST

Item	Kit	Assembled	Delivery
2KRO EPROM module	$ 50.	$ 75.	2 weeks ARO
3P+S I/O module	125.	165.	3 weeks ARO
4KRA-2 RAM module w/2048 8-bit words	135.	185.	2 weeks ARO
4KRA-4 w/4096 8-bit words of RAM	215.	280.	2 weeks ARO
RAM only, AMD 91L02A 500n sec low power	8/$40	–	2 weeks ARO
MB-1 Mother Board	35.	–	2 weeks ARO
VDM-1 Video Display module	160.	225.	Sept. 29, '75 then 3 weeks ARO

Send for our FREE flyer for more complete specifications and for pricing on additional items.

TERMS: All items postpaid if full payment accompanies order. COD orders must include 25% deposit. MasterCharge gladly accepted, but please send us an order with your signature on it.
DISCOUNTS: Orders over $375 may subtract 5%; orders over $600 may subtract 10%.

Processor Technology
2465 Fourth Street
Berkeley, Ca. 94710 (415) 549-0857

Television

Anyone with a bunch of memory circuits, control logic and a wire wrap gun can whip up a digital video generator with TTL output levels. The problem as I see it is to get that digital video signal into a form that the TV set can digest. The care and feeding of digital inputs to the TV set is the subject of Don Lancaster's contribution to BYTE 2 – an excerpt from his forthcoming book, *TV Typewriter Cookbook,* to be published by Howard W. Sams, Indianapolis, Indiana.

...CARL

by
Don Lancaster

We can get between a TV typewriter and a television style display system either by an rf modulator or a direct video method.

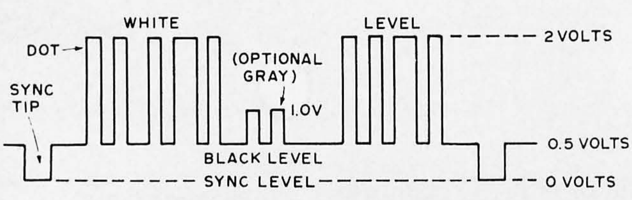
In the rf modulator method, we build a miniature, low power, direct wired TV transmitter that clips onto the antenna terminals of the TV set. This has the big advantage of letting you use any old TV set and ending up with an essentially free display that can be used just about anywhere. No set modifications are needed, and you have the additional advantage of automatic safety isolation and freedom from hot chassis shock problems.

There are two major restrictions to the rf modulator method. The first of these is that transmitters of this type must meet certain exactly spelled out FCC regulations and that system type approval is required. The second limitation is one of bandwidth. The best you can possibly hope for is 3.5 MHz for black and white and only 3 MHz for color, and many economy sets will provide far less. Thus, long character line lengths, sharp characters, and premium (lots of dots) character generators simply aren't compatible with clip-on rf entry.

In the direct video method, we enter the TV set immediately following its video detector but before sync is picked off. A few premium TV sets and all monitors already have a video input directly available, but these are still expensive and rare. Thus, you usually have to modify your TV set, either adding a video input and a selector switch or else dedicating the set to exclusive TV typewriter use. Direct video eliminates the bandwidth restrictions provided by the tuner, i-f strip, and video detector filter. Response can be further extended by removing or shorting the 4.5 MHz sound trap and by other modifications to provide us with longer line lengths and premium characters. No FCC approval is needed, and several sets or monitors are easily driven at once without complicated distribution problems.

There are two limitations to the direct video technique. One is that the set has to be modified to provide direct video entry. A second, and far more severe, restriction, is that many television sets are "hot chassis" or ac-dc sets with one side of their chassis connected to the power line. **These sets introduce a severe shock hazard and cannot be used as TV typewriter video entry displays unless some isolation technique is used with them.** If the TV set has a power transformer, there is usually no hot chassis problem. Transistor television sets and IC sets using no vacuum tubes tend to have power transformers, as do older premium tube type sets. All others (around half the sets around today) do not.

Fig. 1. Standard video interface levels. (Source impedance = 72 or 100 Ohms.)



Direct Video Methods

With either interface approach, we usually start by getting the dot matrix data, blanking, cursor, and sync signals together into one composite video signal whose

Interface

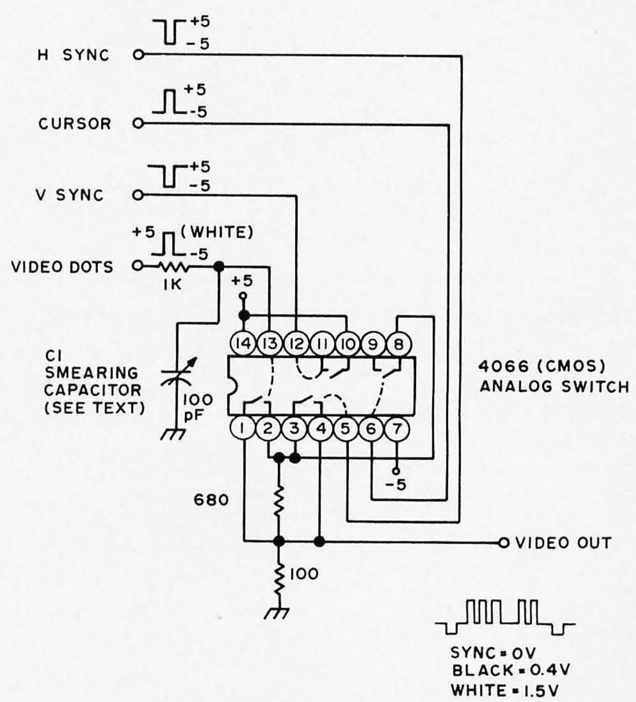
form is useful to monitors and TV sets. A good set of standards is shown in Fig. 1. The signal is dc coupled and always positive going. Sync tips are grounded and blacker than black. The normal open circuit black level is positive by one-half a volt, and the white level is two volts positive. In most TV camera systems, intermediate levels between the half volt black level and the two volt white level will be some shade of gray, proportionately brighter with increasing positive voltage. With most TV typewriter systems, only the three states of zero volts (sync), half a volt (black), and two volts (white dot) would be used. One possible exception would be an additional one volt dot level for a dim but still visible portion of a message or a single word.

The usual video source impedance is either 72 or 100 Ohms. Regardless of how far we travel with a composite video output, some sort of shielding is absolutely essential.

For short runs from board to board or inside equipment, tightly twisted conductors should be OK, as should properly guarded PC runs. Fully shielded cables should be used for interconnections between the TVT and the monitor or TV set, along with other long runs. As long as the total cable capacitance is less than 500 pF or so (this is around 18 feet of RG178-U miniature coax), the receiving end of the cable need not be terminated in a 72 or 100 Ohm resistor. When terminated cable systems are in use for long line runs or multiple outputs, they should be arranged to deliver the signal levels of Fig. 1 at their output under termination. Generally, terminated cable systems should be avoided as they need extra in the way of drivers and supply power.

The exact width of the horizontal and vertical sync pulses isn't usually too important, so long as the shape and risetime of these pulses are independent of position control settings and power supply variations. One exception to this is when you're using a color receiver and a color display. Here, the horizontal sync pulse should be held closely to 5.1 microseconds, so the receiver's color burst sampling does in fact intercept a valid color burst. More on this later.

Fig. 2. Analog switch combiner generates composite video.



Intentional Smear

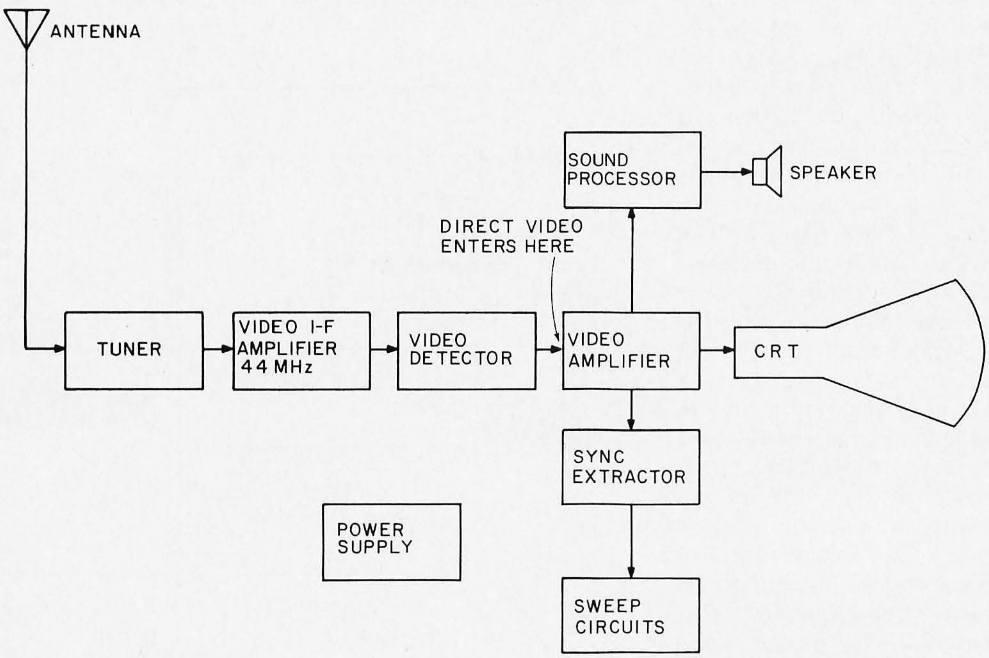
Fig. 2 shows us a typical composite video driver using a 4066 quad analog switch. It gives us a 100 Ohm output impedance and the proper signal levels. Capacitor C1 is used to purposely reduce the video rise and fall times. It is called a smearing capacitor.

Why would we want to further reduce the bandwidth and response of a TV system that's already hurting to begin with? In the case of a quality video monitor, we wouldn't. But if we're using an ordinary run-of-the-mill TV set, particularly one using rf entry, this capacitor can very much improve the display legibility and contrast. Why?

Because we are interested in getting the most legible character of the highest contrast we can. This is not necessarily the one having the sharpest dot rise and fall times. Many things interact to determine the upper video response of a TV display. These include the tuner settings and the i-f response and alignment, the video detector response, video peaking, the sound trap setting, rf cable reflections, and a host of other responses. Many of these stages are underdamped and will ring if fed too sharp a risetime input, giving us a ghosted, shabby, or washed out character. By reducing the video bandwidth going into the system, we can move the dot matrix energy lower in frequency, resulting in cleaner characters of higher contrast.

For most TV displays, intentional smearing will help the contrast, legibility, and overall appearance. The ultimate limit to this occurs when the dots overlap and become illegible. The

Fig. 3. Block diagram of typical B and W television.



optimum amount of intentional smear is usually the value of capacitance that is needed to just close the inside of a "W" presented to the display.

Adding a Video Input

Video inputs are easy to add to the average television set, provided you follow some reasonable cautions. First and foremost, you **must** have an accurate and complete schematic of the set to be modified, preferably a Sams Photofact or something similar. The first thing to check is the power supply on the set. If it has a power transformer and has the chassis properly safety isolated from the power line, it's a good choice for a TVT monitor. This is particularly true of recent small screen, solid state portable TV sets. On the other hand, if you have a hot chassis type with one side of the power line connected to the chassis, you should avoid its use if at all possible. If you must use this type of set, be absolutely certain to use one of the safety techniques outlined later in Fig. 8.

A block diagram of a typical TV set appears in Fig. 3. UHF or VHF signals picked up by the tuner are downconverted in frequency to a video i-f frequency of 44 MHz and then filtered and amplified. The output of the video i-f is transformer coupled to a video detector, most often a small signal germanium diode. The video detector output is filtered to remove the carrier and then routed to a video amplifier made up of one or more tubes or transistors.

At some point in the video amplification, the black and white signal is split three ways. First, a reduced bandwidth output routes sync pulses to the sync separator stage to lock the set's horizontal and vertical scanning to the video. A second bandpass output sharply filtered to 4.5 MHz extracts the FM sound subcarrier and routes this to a sound i-f amplifier for further processing. The third output is video, which is strongly amplified and then capacitively coupled to the cathode of the picture tube.

The gain of the video amplifier sets the contrast of the display, while the bias setting on the cathode of the picture tube (with respect to its *grounded control grid*) sets the display brightness. Somewhere in the video amplifier, further rejection of the 4.5 MHz sound subcarrier is usually picked up to minimize picture interference. This is called a sound trap. Sound traps can be a series resonant circuit to ground, a parallel resonant circuit in the video signal path, or simply part of the transformer that is picking off the sound for more processing.

The video detector output is usually around 2 volts peak to peak and usually subtracts from a white level bias setting. The stronger the signal, the more negative the swing, and the blacker the picture. Sync tips are blacker than black, helping to blank the display during retrace times.

Fig. 4 shows us the typical video circuitry of a transistor black and white television. Our basic circuit consists of a diode detector, a unity gain emitter follower, and a variable gain video output stage that is capacitively coupled to the picture tube. The cathode bias sets the brightness, while the video gain sets the contrast. Amplified signals for sync and sound are removed from the collector of the video driver by way of a 4.5 MHz resonant transformer for the sound and a low pass filter for the sync. A parallel resonant trap set to 4.5 MHz eliminates sound interference. Peaking coils on each stage extend the bandwidth by providing higher impedances and thus higher gain to high frequency video signals.

Note particularly the biasing of the video driver. A bias network provides us with a stable source of 3 volts. In the absence of input video, this 3 volts sets the white level of the display, as well as establishing proper bias for both stages. As an increasing signal appears at the last video output transformer, it is negatively rectified by the video detector, thus lowering the 3 volts proportionately. The stronger the signal, the blacker the picture. Sync will be the strongest of all, giving us a blacker than black bias level of only one volt.

The base of our video driver has the right sensitivity we need for video entry, accepting a maximum of a 2 volt peak to peak signal. It also has the right polarity, for a positive going bias level means a whiter picture. *But, an unmodified set is already biased to the white level, and if we want to enter our own video, this bias must be shifted to the black level.*

We have a choice in any TV of direct or ac coupling of our input video. Direct coupling is almost always better as it eliminates any

Fig. 4. Typical video circuitry of transistor B and W TV set.

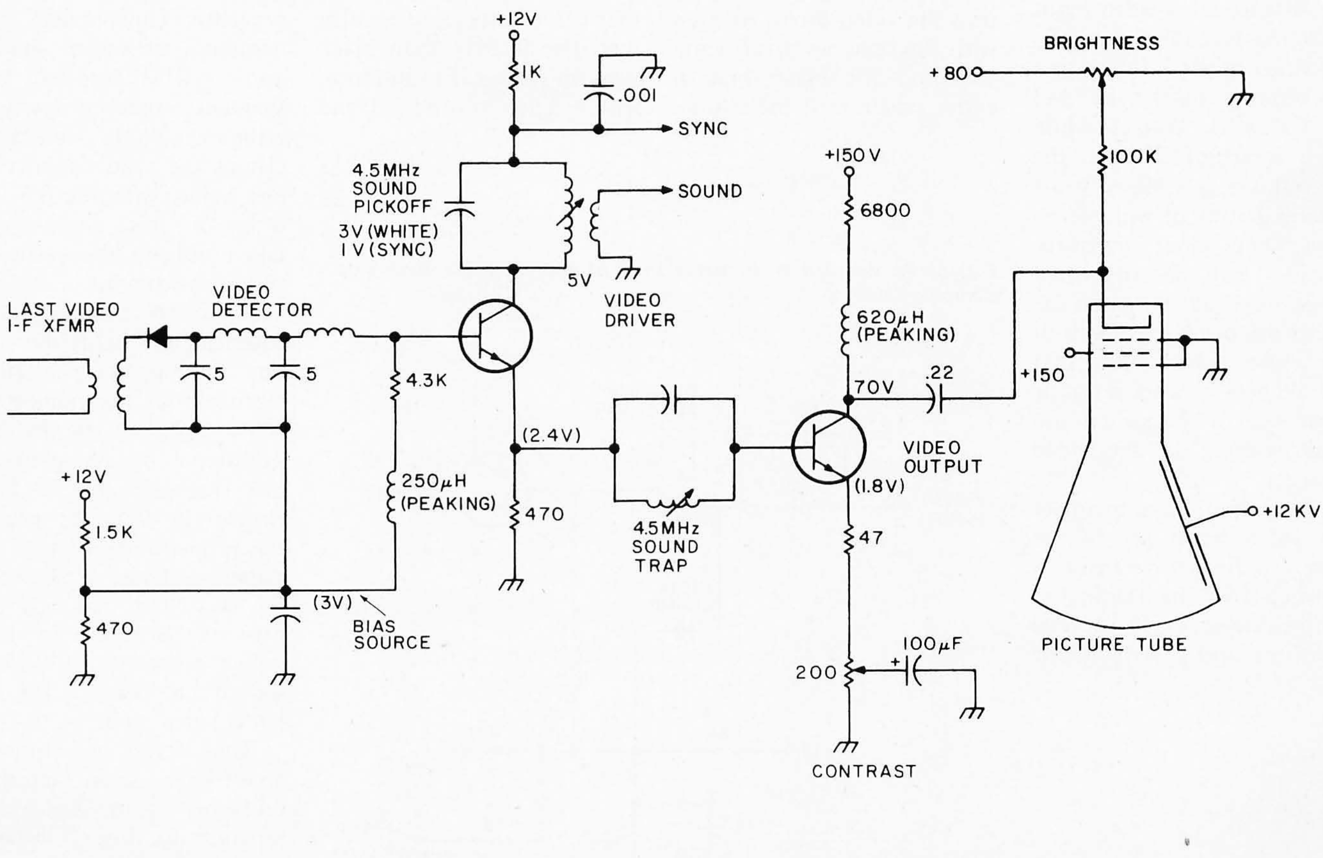
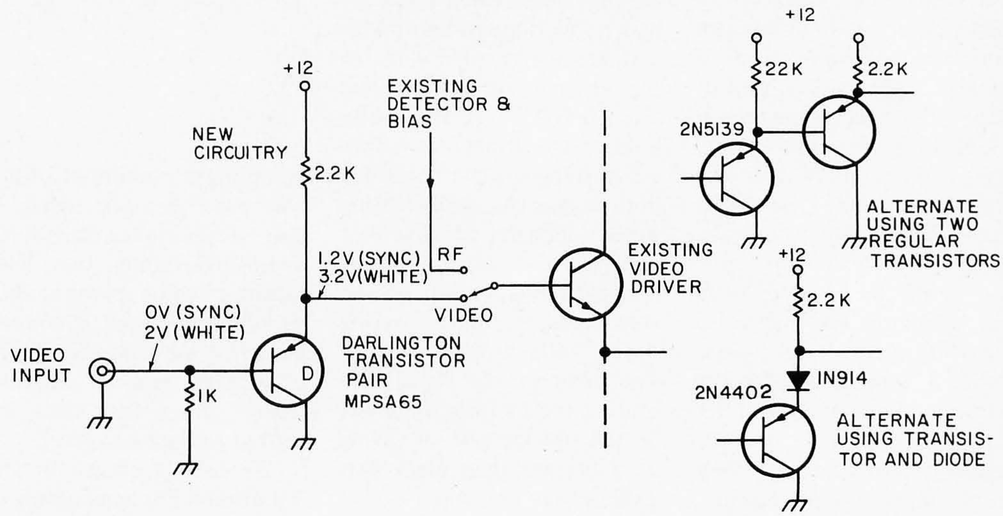


Fig. 5. Direct coupled video uses 1.2 volt offset of Darlington transistor as bias.



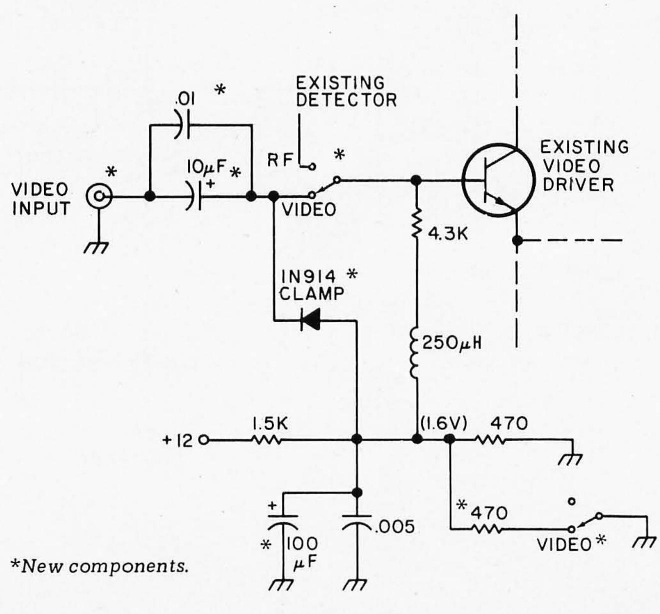
shading effects or any change of background level as additional characters are added to the screen. Fig. 5 shows how we can direct couple our video into a transistor black and white set. We provide a video input, usually a BNC or a phono jack, and route this to a PNP Darlington transistor or transistor pair, borrowing around 5 mils from the set's +12 volt supply. This output is routed to the existing video driver stage through a SPDT switch that either picks the video input or the existing video detector and bias network.

The two base-emitter diode drops in our Darlington transistor add up to a 1.2 volt positive going offset; so, in the absence of a video input or at the base of a sync tip, the video driver is biased to a blacker than black sync level of 1.2 volts. With a white video input of 2 volts, the video driver gets biased to its usual 3.2 volts of white level. Thus, our input transistor provides just the amount of offset we need to match the white and black bias levels of our video driver. Note that the old bias network is on the other side of the switch and does nothing in the video position.

Two other ways to offset our video input are to use two ordinary transistors connected in the Darlington configuration, or to use one transistor and a series diode to pick up the same amount of offset, as shown in Fig. 5. If more or less offset is needed, diodes or transistors can be stacked up further to pick up the right amount of offset.

The important thing is that the video driver ends up with the same level for white bias and for black bias in either position of the switch.

Fig. 6. Ac coupled video needs shift of bias to black level plus a clamping diode.

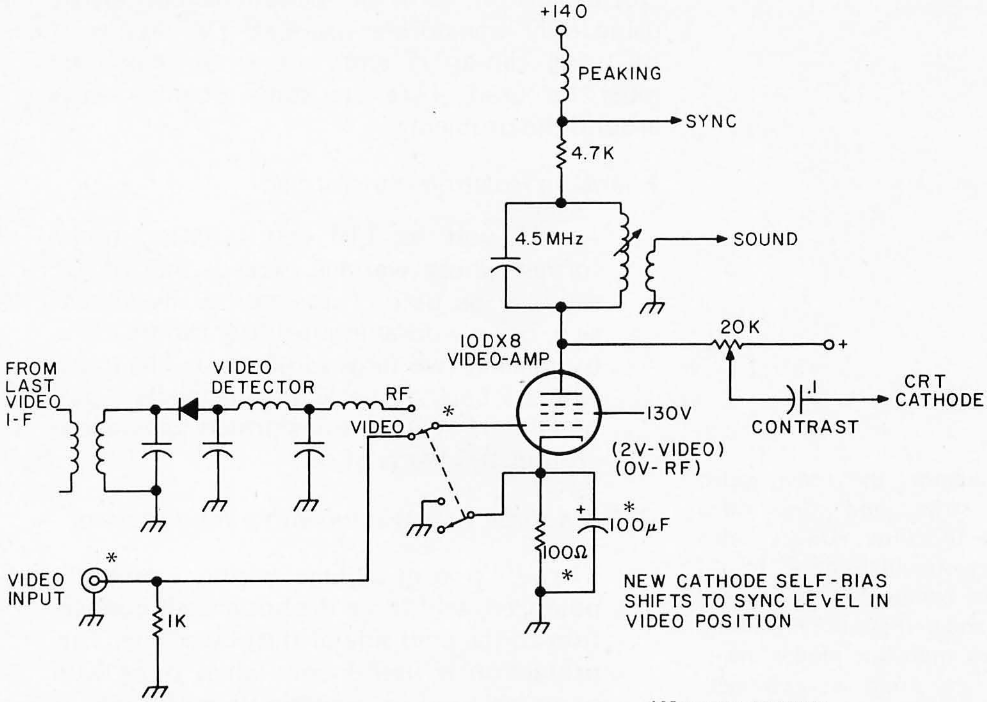


*New components.

Ac or capacitively coupled video inputs should be avoided. Fig. 6 shows a typical circuit. The TV's existing bias network is lowered in voltage by adding a new parallel resistor to ground to give us a voltage that is 0.6 volts more positive than the blacker than black sync tip voltage. For instance, with a 3 volt white level, and 2 volt peak to peak video, the sync tip voltage would be 1 volt; the optimum bias is then 1.6 volts. Input video is capacitively coupled by a fairly large electrolytic capacitor in parallel with a good high frequency capacitor. This provides for a minimum of screen shading and still couples high frequency signals properly. A clamping diode constantly clamps the sync tips to their bias value, with the 0.6 volt drop of this diode being taken out by the extra 0.6 volts provided for in the bias network. This clamping diode automatically holds the sync tips to their proper value, regardless of the number of white dots in the picture. Additional bypassing of the bias network by a large electrolytic may be needed for proper operation of the clamping diode, as shown in Fig. 6. Note that our bias network is used in both switch positions — its level is shifted as needed for the direct video input.

Tube type sets present about the same interface problems as the solid state versions do. Fig. 7 shows a typical direct coupled tube interface. In the unmodified

Fig. 7. Direct coupled video added to tube type B and W television.



**New components.*

circuit, the white level is zero volts and the sync tip black level is minus two volts. If we can find a negative supply (scarce in tube type circuits), we could offset our video in the negative direction by two volts to meet these bias levels.

Instead of this, it is usually possible to self bias the video amplifier to a cathode voltage of +2 volts. This is done by breaking the cathode to ground connection and adding a small resistor (50 to 100 Ohms) between cathode and ground to get a cathode voltage of +2 volts. Once this value is found, a heavy electrolytic bypass of 100 microfarads or more is placed in parallel with the resistor. Switching then grounds the cathode in the normal rf mode and makes it +2 volts in the video entry mode.

In the direct video mode, a sync tip grounded input presents zero volts to the grid, which is self biased minus two volts with respect to the cathode. A white level presents +2 volts to the grid, which equals zero volts grid to cathode.

Should there already be a self bias network on the cathode, it is increased in value as needed to get the black rather than white level bias in the direct video mode.

Hot Chassis Problems

There is usually no shock hazard when we use clip-on rf entry or when we use a direct video jack on a transformer-powered TV. A very severe shock hazard can exist if we use direct video entry with a TV set having one side of the power line connected to the chassis. Depending on which way the line cord is plugged in, there is a 50-50 chance of the hot side of the power line being connected directly to the chassis.

Hot chassis sets, particularly older, power hungry tube versions, should be avoided entirely for direct video entry. If one absolutely must be used, some of the suggestions of Fig. 8 may ease the hazard. These include using an isolation transformer, husky back-to-back filament transformers, three wire power systems, optical coupling of the video input,

and total package isolation. Far and away the best route is simply never to attempt direct video entry onto a hot chassis TV.

Making the Conversion

Fig. 9 sums up how we modify a TV for direct video entry. Always have a complete schematic on hand, and use a transformer style TV set if at all possible. Late models, small screen, medium to high quality solid state sets are often the best display choice. Avoid using junk sets, particularly very old ones. Direct coupling of video is far preferable to ac capacitor coupling. Either method has to maintain the black and white bias levels on the first video amplifier stage. A shift of the first stage quiescent bias from normally white to normally black is also a must. Use short, shielded leads between the video input jack and the rest of the circuit. If a changeover switch is used, keep it as close to the rest of the video circuitry as you possibly can.

Extending Video and Display Bandwidth

By using the direct video input route, we eliminate any bandwidth and response restrictions of an rf modulator, the tuner, video i-f strip, and the video detector filter. Direct video entry should bring us to a 3 MHz bandwidth for a color set and perhaps 3.5 MHz for a black and white model, unless we are using an extremely bad set. The resultant 6 to 7 million dot per second rate is adequate for short character lines of 32, 40, and possibly 48 characters per line. But the characters will smear and be illegible if we try to use longer line lengths and premium (lots of dots) character generators on an ordinary TV. Is there anything we can do to the set to extend the video bandwidth and display response for these longer line lengths?

In the case of a color TV, the answer is probably no. The video response of a color set is limited by an essential delay line and an essential 3.58 MHz trap. Even if we were willing to totally separate the chrominance and luminance channels, we'd still be faced with an absolute limit set by the number of holes per horizontal line in the shadow mask of the tube. This explains why video color displays are so expensive and so rare. Later on, we'll look at what's involved in adding color to the shorter line lengths.

With a black and white TV, there is often quite a bit

Fig. 8. Getting Around a Hot Chassis Problem.

Hot chassis problems can be avoided entirely by using only transformer-powered TV circuits or by using clip-on rf entry. If a hot chassis set must be used, here are some possible ways around the problem:

1. Add an isolation transformer.

A 110 volt to 110 volt isolation transformer whose wattage exceeds that of the set may be used. These are usually expensive, but a workable substitute can be made by placing two large surplus filament transformers back to back. For instance, a pair of 24 volt, 4 Amp transformers can handle around 100 Watts of set.

2. Use a three wire system with a solid ground.

Three prong plug wiring, properly polarized, will force the hot chassis connection to the cold side of the power line. This protection is useful only when three wire plugs are used in properly wired outlets. A severe shock hazard is reintroduced if a user elects to use an adaptor or plugs the system into an unknown or improperly wired outlet. The three wire system should **NOT** be used if anyone but yourself is **ever** to use the system.

3. Optically couple the input video.

Light emitting diode-photocell pairs are low in cost and can be used to optically couple direct video, completely isolating the video input from the hot chassis. Most of these optoelectronic couplers do not have enough bandwidth for direct video use; the Litronix IL-100 is one exception. Probably the simplest route is to use two separate opto-isolators, one for video and one for sync, and then recombine the signals inside the TV on the hot side of the circuit.

4. Use a totally packaged and sealed system.

If you are only interested in displaying messages and have no other input/output devices, you can run the entire circuit hot chassis, provided everything is sealed inside one case and has no chassis-to-people access. Interface to teletypes, cassettes, etc., cannot be done without additional isolation, and servicing the circuit presents the same shock hazards that servicing a hot chassis TV does.

we can do to present long lines of characters, depending on what set you start out with and how much you are willing to modify the set.

The best test signal you can use for bandwidth extension is the dot matrix data you actually want to display, for the frequency response, time delay, ringing, and overshoot all get into the act. What we want to end up with is a combination that gives us reasonably legible characters.

A good oscilloscope (15 MHz or better bandwidth) is very useful during bandwidth extension to show where the signal loses its response in the circuit. At any time during the modification process, there is usually one response bottleneck. This, of course, is what should be attacked first. Obviously the better a TV you start with, the easier will be the task. Tube type gutless wonders, particularly older ones, will be much more difficult to work with than with a modern, small screen, quality solid state portable.

Several of the things we can do are watching the control settings, getting rid of the sound trap, minimizing circuit strays, optimizing spot size, controlling peaking, and shifting to higher current operation. Let's take a look at these in turn.

Control Settings

Always run a data display at the lowest possible contrast and using only as much brightness as you really need. In many circuits, low contrast means a lower video amplifier gain, and thus less of a gain-bandwidth restriction.

Eliminate the Sound Trap

The sound trap adds a notch at 4.5 MHz to the video response. If it is eliminated or switched out of the circuit, a wider video bandwidth automatically

Fig. 9. How to Add a Direct Video Input to a TV Set.

1. Get an accurate and complete schematic of the set — either from the manufacturer's service data or a Photofact set. **Do not try adding an input without this schematic!**

2. Check the power supply to see if a power transformer is used. If it is, there will be no shock hazard, and the set is probably a good choice for direct video use. If the set has one side of the power line connected to the chassis, a severe shock hazard exists, and one of the techniques of Fig. 8 should be used. **Avoid the use of hot chassis sets.**

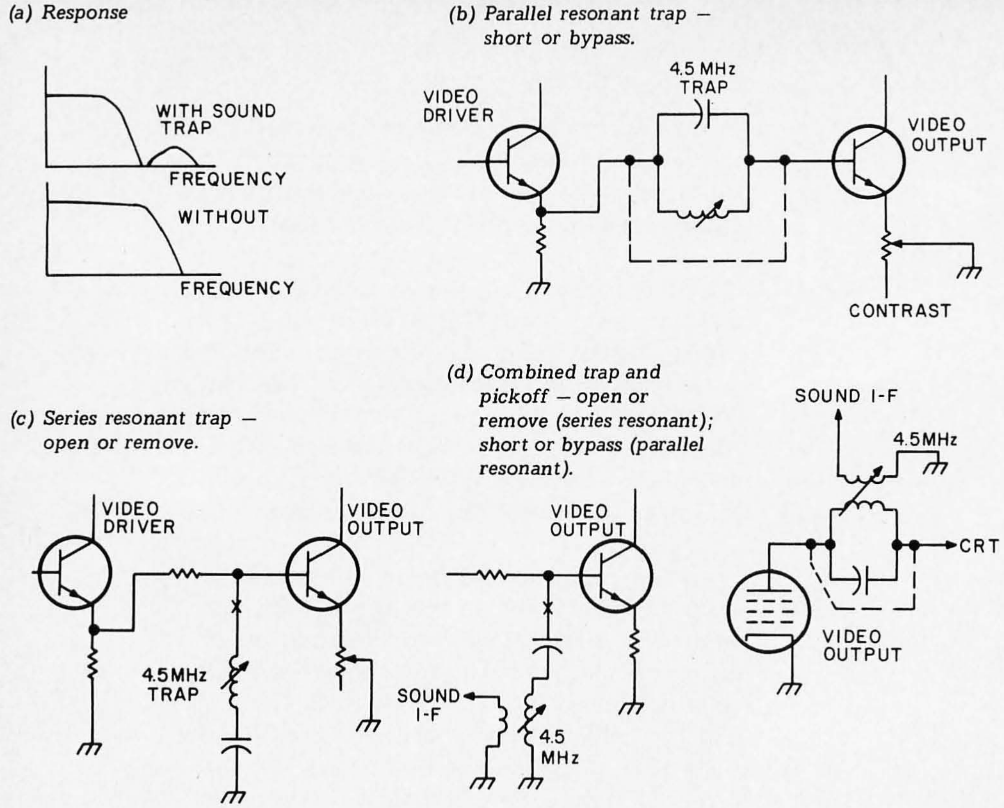
3. Find the input to the first video amplifier stage. Find out what the white level and sync level bias voltages are. The marked or quiescent voltage is usually the white level; sync is usually 2 volts less. A transistor TV will typically have a +3 volt white level and a +1 volt sync level. A tube type TV will typically have a zero volt white level and a -2 volt sync level.

4. Add a changeover switch using minimum possible lead lengths. Add an input connector, either a phono jack or the premium BNC type connector. Use shielded lead for interconnections exceeding three inches in length.

5. Select a circuit that couples the video and biases the first video amplifier stage so that the white and sync levels are preserved. For transistor sets, the direct coupled circuits of Fig. 5 may be used. For tube sets, the circuit of Fig. 7 is recommended. Avoid the use of ac coupled video inputs as they may introduce shading problems and changes of background as the screen is filled.

6. Check the operation. If problems with contrast or sync tearing crop up, recheck and adjust the white and sync input levels to match what the set uses during normal rf operation. Note that the first video stage must be biased to the **white** level during rf operation and to the **sync** level for direct video use. The white level is normally two volts more positive than the sync level.

Fig. 10. Removing the sound trap can extend video bandwidth.



results. Fig. 10 shows us the response changes and the several positions for this trap. Generally, series resonant traps are opened and parallel resonant traps are shorted or bypassed through suitable switching or outright elimination. The trap has to go back into the circuit if the set is ever again used for ordinary program reception. Sometimes simply backing the slug on the trap all the way out will improve things enough to be useful.

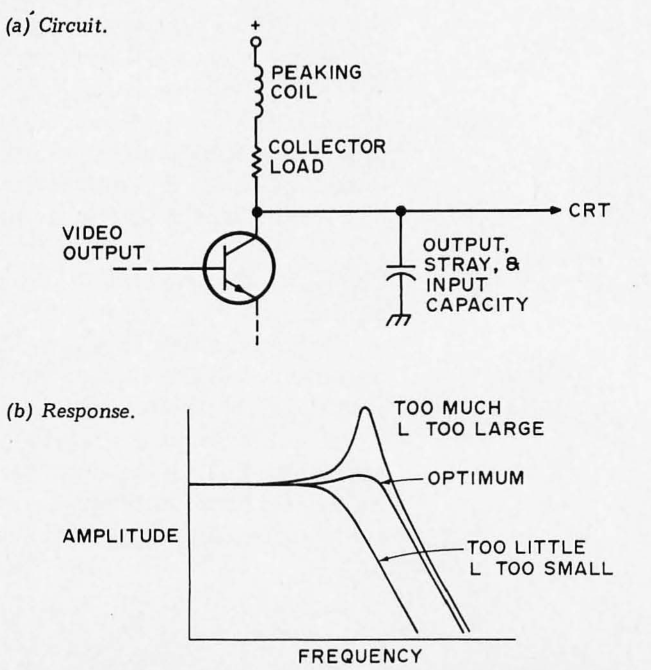
Minimizing Strays

One of the limits of the video bandwidth is the stray capacitance both inside the video output stage and in the external circuitry. If the contrast control is directly in the signal path and if it has long leads going to it, it may be hurting the response. If you are using the TV set exclusively for data display, can you rearrange the control location and simplify and shorten the video output to picture tube interconnections?

Additional Peaking

Most TV sets have two peaking networks. The first of these is at the video detector output and compensates for the vestigial sideband transmission signal that makes sync and other low frequency signals double the amplitude of the higher frequency ones. The second of these goes to the collector or plate of the video output stage and raises the circuit impedance and thus the effective gain for very high frequencies. Sometimes you can alter this second network to favor dot presentations. Fig. 11 shows a typical peaking network and the effects of too little or too much peaking. Note that the stray capacitance also enters into the peaking, along with the video amplifier output capacitance and the picture tube's input capacitance. Generally, too little peaking will give you low contrast dots, while too much will give you sharp dots, but will run dots together and shift the more continuous portions of the characters objectionably. Peaking is changed by increasing or decreasing the series inductor from its design value.

Fig. 11. Adjusting the peaking coil can extend video response.



Running Hot

Sometimes increasing the operating current of the video output stage can increase the system bandwidth – IF this stage is in fact the limiting response, IF the power supply can handle the extra current, IF the stage isn't already parked at its gain-bandwidth peak, and IF the extra heat can be gotten rid of without burning anything up. Usually, you can try adding a resistor three times the plate or collector load resistor in parallel, and see if it increases bandwidth by 1/3. Generally, the higher the current, the wider the bandwidth, but watch

carefully any dissipation limits. Be sure to provide extra ventilation and additional heatsinking, and check the power supply for unhappiness as well. For major changes in operating current, the emitter resistors and other biasing components should also be proportionately reduced in value.

Spot Size

Even with excellent video bandwidth, if you have an out-of-focus, blooming, or changing spot size, it can completely mask character sharpness. Spot size ends up the ultimate limit to resolution, regardless of video bandwidth.

Once again, brightness and contrast settings will have a profound effect, with too much of either blooming the spot. Most sets have a focus jumper in which ground or a positive voltage is selected. You can try intermediate values of voltage for maximum sharpness. Extra power supply filtering can sometimes minimize hum and noise modulation of the spot.

Anything that externally raises display contrast will let you run with a smaller beam current and a sharper spot. Using circularly polarized filters, graticule masks, or simple colored filters can minimize display washout from ambient lighting. Fig. 12 lists several sources of material for contrast improvement. Much of this is rather expensive, with pricing from \$10 to \$25 per square foot being typical. Simply adding a hood and positioning the display away from room lighting will also help and is obviously much cheaper.

Direct Rf Entry

If we want the convenience of a "free" display, the freedom from hot chassis problems, and "use it anywhere" ability, direct rf entry is the obvious choice. Its two big limitations are the need for FCC type approval, and a limited video bandwidth that in turn limits the number of characters per line and the number of dots per character.

An rf interface standard is shown in Fig. 13. It consists of an amplitude modulated carrier of one of the standard television channel video frequencies of Fig. 14. Channel 2 is most often used with a 55.250 MHz carrier frequency, except in areas where a local commercial Channel 2 broadcast is intolerably strong. Circuit cost, filtering problems, and stability problems tend to increase with increasing channel number.

The sync tips are the strongest part of the signal, representing 100% modulation, often something around 4 millivolts rms across a 300 Ohm line. The black level is 75% of the sync level, or about 3 millivolts for 4 millivolt sync tips. White level is less than 10% of maximum. Note that the signal is weakest when white and strongest when sync. This is the exact opposite of the video interface of Fig. 1.

Rf modulators suitable for clip-on rf entry TV typewriter use are called Class 1 TV Devices by the FCC. A Class 1 TV device is supposed to meet the rules and regulations summarized in Fig. 15.

Fig. 16 shows us a block diagram of the essential parts of a TV modulator. We start

Fig. 12. Contrast Enhancing Filter Materials.

Circularly polarized filters:

Polaroid Corp.
Cambridge MA 02139

Anti-reflection filters:

Panelgraphic Corp.
10 Henderson Dr.
West Caldwell NJ 07006

Light control film:

3M Visual Products Div.
3M Center
St. Paul MN 55101

Acrylic plexiglas filter sheets:

Rohm and Haas
Philadelphia PA 19105

Fig. 13. Standard rf interface levels. Impedance = 300Ω. Carrier frequency per Fig. 14.

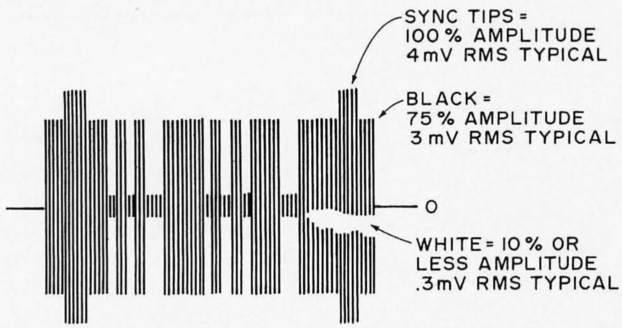


Fig. 14. Television Picture Carrier Frequencies.

Channel 2	55.25 MHz
Channel 3	61.25 MHz
Channel 4	67.25 MHz
Channel 5	77.25 MHz
Channel 6	83.25 MHz

Fig. 15. FCC Regulations on Class 1 TV Devices. More complete information appears in subpart H of Part 15 and subpart F of Part 2 of the Federal Communications Commission Rules and Regulations. *It is available at many large technical libraries.*

A Class 1 TV device generates a video modulated rf carrier of a standard television channel frequency. It is directly connected to the antenna terminals of the TV set.

The maximum rms rf voltage must be less than 6 millivolts using a 300 Ohm output line.

The maximum rf voltage on any frequency more than 3 MHz away from the operating channel must be more than 30 dB below the peak in-channel output voltage.

An antenna disconnect switch of at least 60 dB attenuation must be provided.

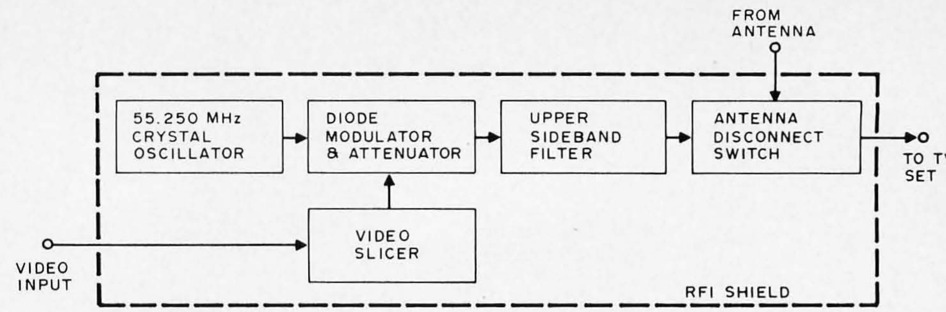
No user adjustments are permitted that would exceed any of the above specifications.

Residual rf radiation from case, leads and cabinet must be less than 15 microvolts per meter.

A Class 1 TV device must not interfere with TV reception.

Type approval of the circuit is required. A filing fee of \$50 and an acceptance fee of \$250 is involved.

Fig. 16. Block diagram of rf modulator.



with a stable oscillator tuned to one of the Fig. 14 frequencies. A crystal oscillator is a good choice, and low cost modules are widely available. The output of this oscillator is then amplitude modulated. This can be done by changing the bias current through a silicon small signal diode. One milliampere of bias current makes the diode show an ac and rf impedance of 26 Ohms. Half a mil will look like 52 Ohms, and so on. The diode acts as a variable resistance attenuator in the rf circuit, whose bias is set and changed by the video circuit.

Since diode modulators are non-linear, we can't simply apply a standard video signal to them and get a standard rf signal out. A differential amplifier circuit called a video slicer may be used to compensate for this non-linearity. The video slicer provides three distinct currents to the diode modulator. One of these is almost zero for the white level, while the other two provide the black and sync levels. A contrast control that sets the slicing level lets you adjust the sync tip height with respect to the black level. The video slicer also minimizes rf getting back into the video. An attenuator to reduce the size of the modulated signal usually follows the diode modulator.

An upper side band filter removes most of the lower sideband from the AM modulated output, giving us a vestigial sideband signal that stays inside the channel band limits. This same filter eliminates second harmonic effects and other spurious noise. The filter's output is usually routed to an antenna disconnect switch and the TV's antenna terminals. A special switch is needed to provide enough isolation.

Some of the actual circuitry involved is shown in Fig. 17. The video slicer consists of a pair of high gain, small signal NPN transistors, while the oscillator is a commercially available module.

Rf entry systems always must be direct coupled to the antenna terminals of the set and should never provide any more rf than is needed for a minimum snow-free picture. They should be permanently tuned to a single TV channel. Under no circumstances should an antenna or cable service hookup remain connected to the set during TVT use, nor should radiation rather than a direct rf cable connection ever be used.

Color Techniques

We can add a full color capability to a TV typewriter system fairly easily and cheaply — provided its usual black and white video dot rate is low enough in frequency to be attractively displayed on an ordinary color TV. Color may be used to emphasize portions of a message, to attract attention, as part of an electronic game, or as obvious added value to a graphics display. Color techniques work best on TV typewriter systems having a horizontal frequency very near 15,735 Hertz.

All we basically have to do is generate a subcarrier sine wave to add to the video output. The phase of this subcarrier (or its time delay) is shifted with respect to what the phase was immediately after each horizontal sync pulse to generate the various colors.

Fig. 18 shows us the differences between normal color and black and white operation. Black and white baseband video is some 4 MHz wide and has a narrow 4.5 MHz sound subcarrier. The video is amplitude modulated, while the sound is narrow band frequency

Fig. 17. Channel two oscillator, modulator, video slicer and attenuator. R sets output level.

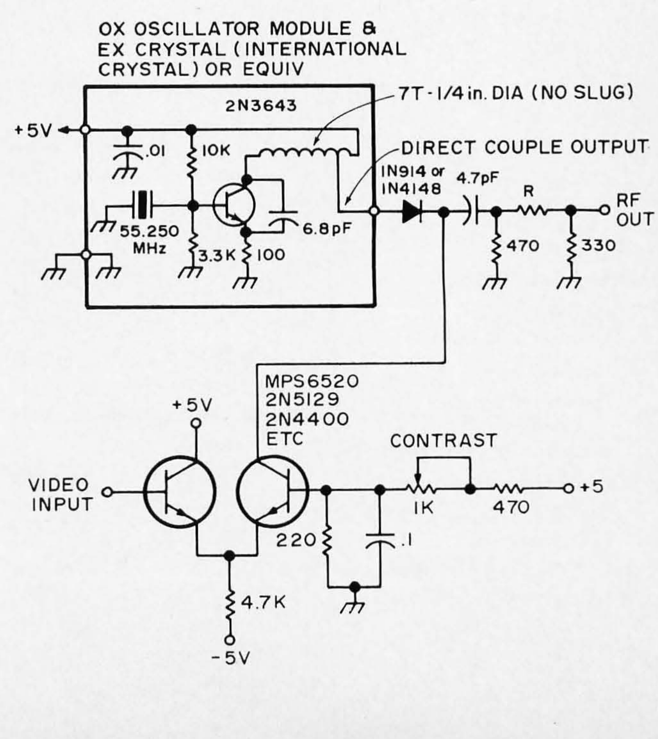
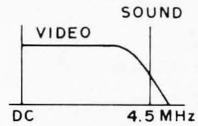
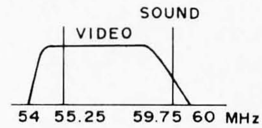


Fig. 18. Differences between color and black and white spectra.

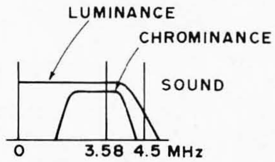
(a) Black and white – baseband video.

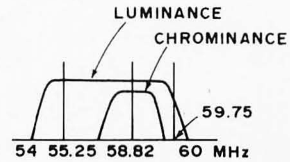




(b) Black and white – Channel two rf.

(c) Color – baseband video.





(d) Color – Channel two rf.

modulated. This translates up to a 6 MHz rf channel with a vestigial lower sideband as shown in Fig. 18(b).

To generate color, we add a new pilot or subcarrier at a magic frequency of 3.579545 MHz – see Fig. 18(c). What was the video is now called the luminance, and is the same as the brightness in a black and white system. The new subcarrier and its modulation is called the chrominance signal and determines what color gets displayed and how saturated the color is to be.

Since the black and white information is a sampled data system that is scanned at the vertical and horizontal rates, there are lots of discrete holes in the video spectrum that aren't used. The color subcarrier is designed to stuff itself into these holes (exactly in a NSTC color system, and pretty much in a TVT display). Both chrominance and luminance signals use the same spectral space, with the one being where the other one isn't, overlapping comb style.

The phase or relative delay of the chrominance signal with respect to a reference determines the instantaneous color, while the amplitude of this signal with respect to the luminance sets the saturation of the color. Low amplitudes generate white or pastel shades, while high amplitudes of the chrominance signal produce saturated and deep colors.

At least eight cycles of a reference or burst color phase are transmitted immediately following each horizontal sync pulse as a timing reference, as shown in Fig. 19. The burst is around 25% of maximum amplitude, or about the peak to peak height of a sync pulse.

The TV set has been trained at the factory to sort all this out. After video detection, the set splits out the chrominance channel with a bandpass amplifier and then synchronously demodulates it with respect to an internal 3.58 MHz reference. The phase of this demodulation sets the color and the amplitude sets the saturation by setting the ratios of electron beam currents on the picture tube's red, blue and green guns.

Meanwhile, the luminance channel gets amplified as brightness style video. It is delayed with a delay line to make up for the time delay involved in the narrower band color processing channel. It is then filtered with two traps – the 4.5 MHz sound trap, and a new trap to get rid of any remaining 3.58 MHz color subcarrier that's left. The luminance output sets the overall brightness by modulating the cathodes of all three color guns simultaneously.

Just after each horizontal sync pulse, the set looks for the reference burst and uses this reference in a phase

Fig. 19 Adding a color reference burst to the back porch of the horizontal sync pulses.

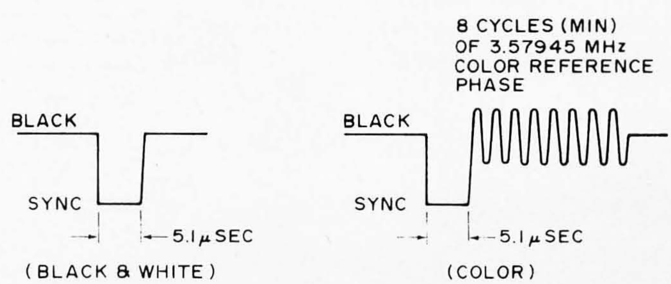


Fig. 20. Colors Are Generated by Delaying or Phase Shifting the Burst Frequency.

Color	Approximate Phase	Approximate Delay
Burst	0°	0
Yellow	15°	12 nanoseconds
Red	75°	58 nanoseconds
Magenta	135°	105 nanoseconds
Blue	195°	151 nanoseconds
Cyan	255°	198 nanoseconds
Green	315°	244 nanoseconds

detector circuit to keep its own 3.58 MHz reference locked to the version being transmitted.

Fig. 20 shows us the phase angles related to each color with respect to the burst phase. It also shows us the equivalent amount of delay we need for a given phase angle. Since we usually want only a few discrete colors, it's far easier to digitally generate colors simply by delaying the reference through gates or buffers, rather than using complex and expensive analog phase shift methods.

Strictly speaking, we should control both the chrominance phase and amplitude to be able to do both pastel and strongly saturated colors. But simply keeping the subcarrier amplitude at the value we used for the burst – around 25% of video amplitude – is far simpler and will usually get us useful results.

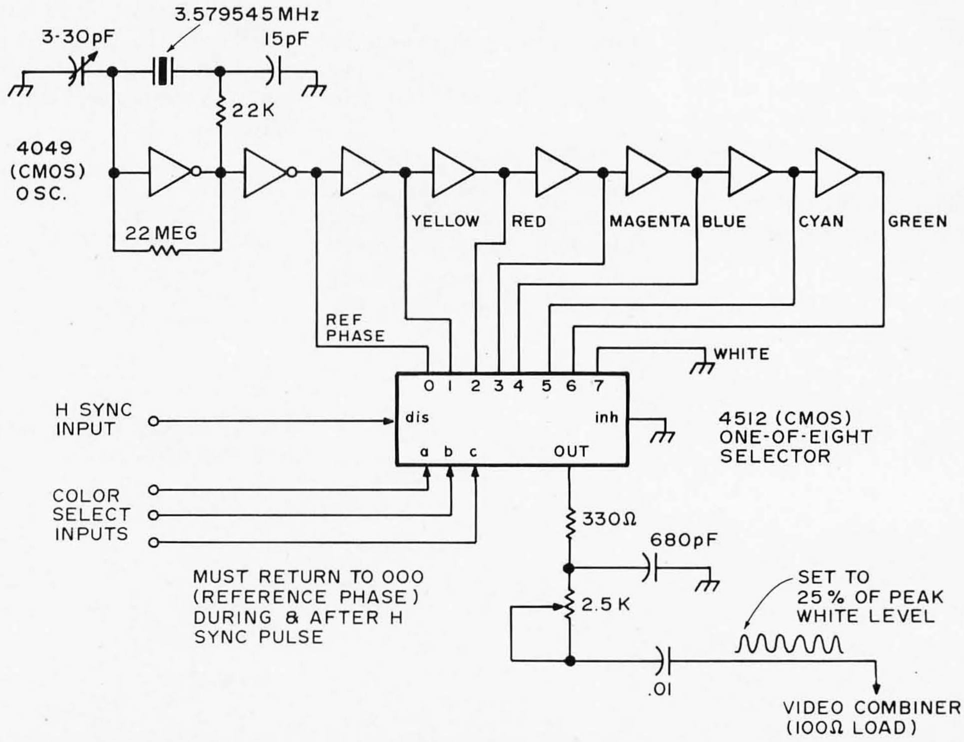
A circuit to add color to a TV typewriter is shown in Fig. 21. A 3.579545 MHz crystal oscillator drives a string of CMOS buffers that make up a digital delay line. The output delays caused by the propagation delay times in each buffer can be used as is, or can be trimmed to specific colors by varying the supply voltage.

The reference phase and the delayed color outputs go to a one-of-eight data selector. The data selector picks either the reference or a selected color in response to a code presented digitally to the three select lines. The logic that is driving this selector must return to the reference phase position (000) immediately before, during and for a minimum of a few microseconds after each horizontal sync pulse. This gives the set a chance to lock and hold onto the reference color burst.

The chrominance output from the data selector should be disabled for the duration of the sync pulses and any time a white screen display is wanted. The output chrominance signal is RC filtered to make it somewhat sinusoidal. It's then cut down in amplitude to around one-quarter the maximum video white level and is capacitively coupled to the 100 Ohm video output of Fig. 2 or otherwise summed into the video or rf modulator circuitry. For truly dramatic color effects, the *amplitude* and delay of the chrominance signal can be changed in a more complex version of the same circuit.

More information useful in solving television interface appears in the *Television Engineering Handbook*, by Donald Fink, and in various issues of the *IEEE Transactions on Consumer Electronics.*

Fig. 21. Color subcarrier generator. Hex buffer used as delay line. Use supply voltage variation on 4050 to trim colors.



The MODULAR MICROS from MARTIN RESEARCH

Here's why the new *MIKE 2* and *MIKE 3* are the best values in microcomputers today!

8008 OR 8080

Martin Research has solved the problem bothering many potential micro users .... whether to go with the economical 8008 microprocessor, or step up to the powerful 8080. Our carefully designed bus structure allows either processor to be used in the same system!

The *MIKE 3* comes with an 8080 CPU board, complete with crystal-controlled system timing. The *MIKE 2* is based on the 8008. To upgrade from an 8008 to an 8080, the user unplugs the 8008 CPU board and plugs in the 8080 CPU. Then he unplugs the 8008 MONITOR PROM, and plugs in the 8080 MONITOR PROM, so that the system recognizes the 8080 instruction set. That's about it!

If the user has invested in slow memory chips, compatible with the 8008 but too slow for the 8080 running at full speed, he will have to make the 8080 wait for memory access—an optional feature on our boards. Better still, a 4K RAM board can be purchased from Martin Research with fast RAM chips, capable of 8080 speeds, at a cost no more that you might expect to pay for much slower devices.

In short, the *MIKE 2* user can feel confident in developing his 8008 system with expanded memory and other features, knowing that his *MIKE 2* can be upgraded to a *MIKE 3*—an 8080 system—in the future.

EASE OF PROGRAMMING

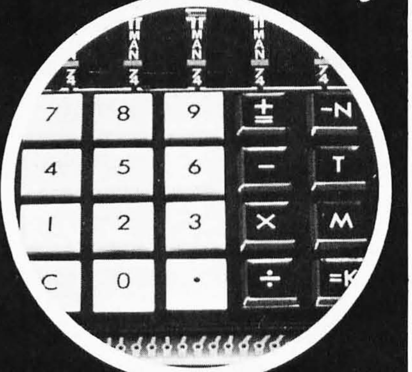
Instructions and data are entered simply by punching the 20-pad keyboard. Information, in convenient octal format, appears automatically on the seven-segment display. This is a pleasant contrast to the cumbersome microcomputers which require the user to handle all information bit-by-bit, with a confusing array of twenty-odd toggle switches and over thirty red lights!

A powerful MONITOR program is included with each microcomputer, stored permanently in PROM memory. The MONITOR continuously scans the keyboard, programming the computer as keys are depressed.

Say the user wishes to enter the number *135* (octal for an 8008 OUTPUT 16 instruction). He types *1,* and the right-hand three digits read *001.* Then he presses *3,* and the digits say *013.* Finally he punches the *5,* and the display reads *135.* Notice how the MONITOR program

(Continued in column 3.)

The MIKE *Family*



Introducing the family of modular micros from Martin Research!

Choose either the economical 8008 processor, or the powerful 8080. Either CPU is compatible with our advanced bus structure! Plus, a convenient monitor program, in PROM memory, allows you to enter instructions with the ease of a handheld calculator. Six large digits display data in octal format.

Modularity makes for easy expansion. First quality parts throughout. Professionally made PC boards with plated holes, soldermask protection. 8080 CPU board features versatile interrupt structure, multiprocessing capability. Easy interfacing to input and output ports.

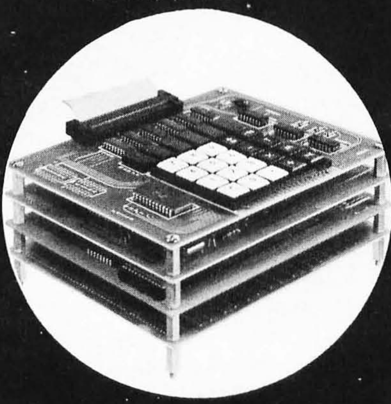
MIKE 303A: CPU board with 8080, keyboard/display board, PROM/RAM board monitor PROM (256 bytes of RAM), breadboard, hardware, and instructions: $395.00 kit, $495.00 assembled and tested.

MIKE 203A: CPU board with 8008, keyboard/display, PROM/RAM, breadboard, hardware, and instructions: $270.00 kit, $345.00 A&T.

MIKE 3-5 or 2-5: 4K RAM board with 450 ns static RAM: $165.00 kit, $190.00 A&T.

FREE CATALOG!

Kits: US & Canada only.
Master Charge accepted.
OEMs: write for quantity prices.



MARTIN RESEARCH
Microcomputer Design
1825 S. Halsted St.
Chicago, IL 60608
(312) 829-6932

shifts each digit left automatically as a new digit is entered! The value on the display is also entered into an internal CPU register, ready for the next operation. Simply by pressing the *write* key, for example, the user loads *135* into memory.

The MONITOR program also allows the user to step through memory, one location at a time (starting anywhere), to check his programming. Plus, the Swap Register Option allows use of the interrupt capabilities of the microprocessor: the MONITOR saves internal register status upon receipt of an interrupt request; when the interrupt routine ends, the main program continues right where it left off.

We invite the reader to compare the programmability of the *MIKE* family of microcomputers to others on the market. Notice that some are sold, as basic units, *without any memory capacity at all.* This means they simply cannot be programmed, until you purchase a memory board as an "accessory." Even then, adding RAM falls far short of a convenient, permanent MONITOR program stored in PROM. Instead, you have to enter your frequently-used subroutines by hand, each and every time you turn the power on.

EASY I/O INTERFACE

The *MIKE* family bus structure has been designed to permit easy addition of input and output ports. A hardware interface to the system generally needs only two chips—one strobe decoder, and one latching device (for output ports) or three-state driving device (for inputs). A new I/O board can be plugged in anywhere on the bus; in fact, all the boards in the micro could be swapped around in any position, without affecting operation. I/O addresses are easy to modify by reconnecting the leads to the strobe decoder (full instructions are provided); this is in marked contrast to the clumsy input multiplexer approach sometimes used.

POWER & HOUSING

The micros described to the left are complete except for a cabinet of your own design, and a power supply. The basic micros require +5 V, 1.4 A, and −9 V, 100 MA. The 4K RAM board requires 5 V, 1 A. A supply providing these voltages, and ±12 V also, will be ready soon.

OPTIONS

A number of useful micro accessories are scheduled for announcement. In addition, the *MIKE 3* and *MIKE 2* may be purchased in configurations ranging from unpopulated cards to complete systems. For details, phone, write, or check the reader service card.

LIFE *Line* 2

by
Carl Helmers
Editor, BYTE

LIFE Line 1 (BYTE #1) presented the general picture of the LIFE program application of your computer. That picture includes the rules of the game, methods of interactively entering graphic data, major software components in verbal description and some of the hardware requirements of the game. In this installment, the discussion turns to some of the program design for the LIFE application. The discussion starts "at the top" (overall program flow) and works down to more detailed levels of design, concentrating upon the "evolution algorithm" which generates new patterns from old patterns.

As in the previous LIFE Line, the goal of the article series is as much to explain and instruct as it is to elaborate upon this one particular system. This article concentrates on the program design process as illustrated by a real LIFE example.

What Is This Process – Designing A Program?

For the readers who are only just now beginning to learn the programming of computers, an elementary acquaintance with some machine's language, a BASIC interpreter, or high level languages would tend to give the impression that programming is fundamentally simple. It is! To write a program which fills a single page of listing – whatever the language or machine architecture involved – is not a tremendously difficult task. When it comes to more complex projects – say 1000 or more words of hand or machine-generated code on your microcomputer – the problem is how to preserve the blissful innocence of simplicity in the face of the worldly forces of complexity.

When you begin to talk about programs more complex than a one page assembly or machine code listing of some specialized service routine or simple "gimmick" program (see the Kluge Harp article in this issue), the complexities and subtleties of scale begin to enter into the programming art. For an application such as the LIFE program, proceeding from the vague notion "I want this application" to a working program can be done in innumerable ways – many of which will work quite well. This is the first ambiguity of scale – where do you head as you start programming? Unless you have a unique parallel processing mind, you can't possibly concentrate on the whole problem of programming at once.

In order to make a big application program work, you have to select "bits and pieces" of the desired result, figure out what they do and how they fit into the big picture, then program them one by one. These little pieces of the program – its "modules" – are like the multiple layers of stone blocks in a pyramid. In fact, defining what to do is very much like the tip of some Egyptian tyrant's tomb in the spring flood . . . as the murky generalities recede, more and more of the structure of the program is defined and clarified. Fig. 1 illustrates the pyramid of abstractions at the start of a program design process. The top layer is clear – a LIFE program is the desired goal. The next layer down is for the most part visible through the obscuring water. But the details of the base of the pyramid – while you know they have to be there in some form – are not at all visible at the start. The design process moves the logical "water level" surrounding the pyramid lower and lower as you figure out more and more of the detail content of the program.

Start at the Top . . .

In LIFE Line 1, I mentioned two major functions which compose a practical LIFE program – data entry and manipulation is one, the LIFE evolution algorithm is the second. Together, these functions define the "program control" layer of the LIFE pyramid. Fig. 2 is a flow chart illustrating the program control algorithm which is the top level of the program structure. Although the diagram – and the algorithm – are extremely simple, they

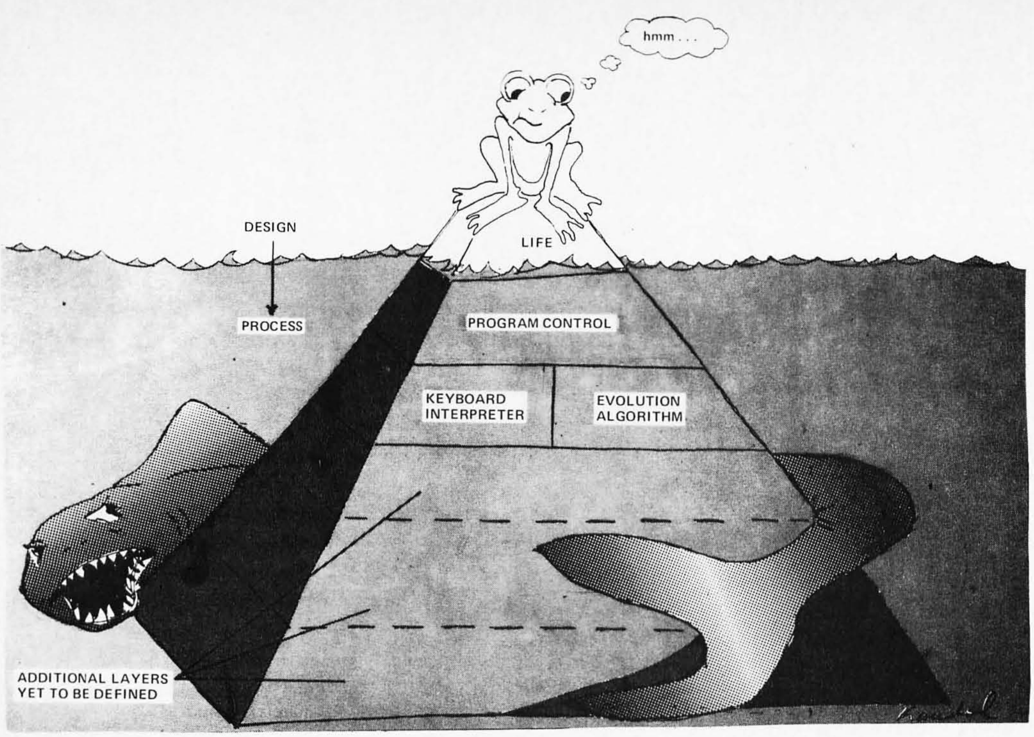


Fig. 1. Defining what to do is like the tip of some Egyptian tyrant's tomb in the spring flood . . . as the murky generalities recede more and more of the structure is defined and clarified . . .

serve a very useful purpose in the program design process: *This high level design has split most of the programming work into two moderately large segments, each of which is less complicated than the whole program.* This view of the problem now gives us two major components upon which to concentrate attention once the top level routine is completed. The program control algorithm of Fig. 2, elaborated in Fig. 3, is the "mortar" which cements together these two blocks of function.

The LIFE program is entered by one of a number of methods. Fig. 2 illustrates branch or jump possibilities from a systems program called a "monitor," "executive" or "operating system" – the preferred way once you get such a system generated. If your system runs "bare bones" with little system-resident software, you might select the starting point and activate the program through use of hardware restart mechanisms and a front panel console.

The first module of the LIFE application to be entered is the KEYBOARD_INTERPRETER, a set of routines which is used to define the initial content of the LIFE grid using interactive commands and the scope display output. The KEYBOARD_INTERPRETER eventually will receive a "GO" command or an "END" command from the user – whereupon it will return to the main routine with the parameters "DONE" and "N" defined. If "N" is greater than zero, control flows to the evolution process – and "N" generations of LIFE will be computed and displayed as they are completed. After the "N" generations have been completed, the scope display and the LIFE grid have the last completed results. If the program is not "DONE," control flows back to the KEYBOARD_INTERPRETER for modification of the data, clearing the screen and starting over, or other operations. If the program is "DONE" then the control flows back to the systems programs – or to a halt point.

This program control algorithm is elaborated in more explicit detail in Fig. 3.

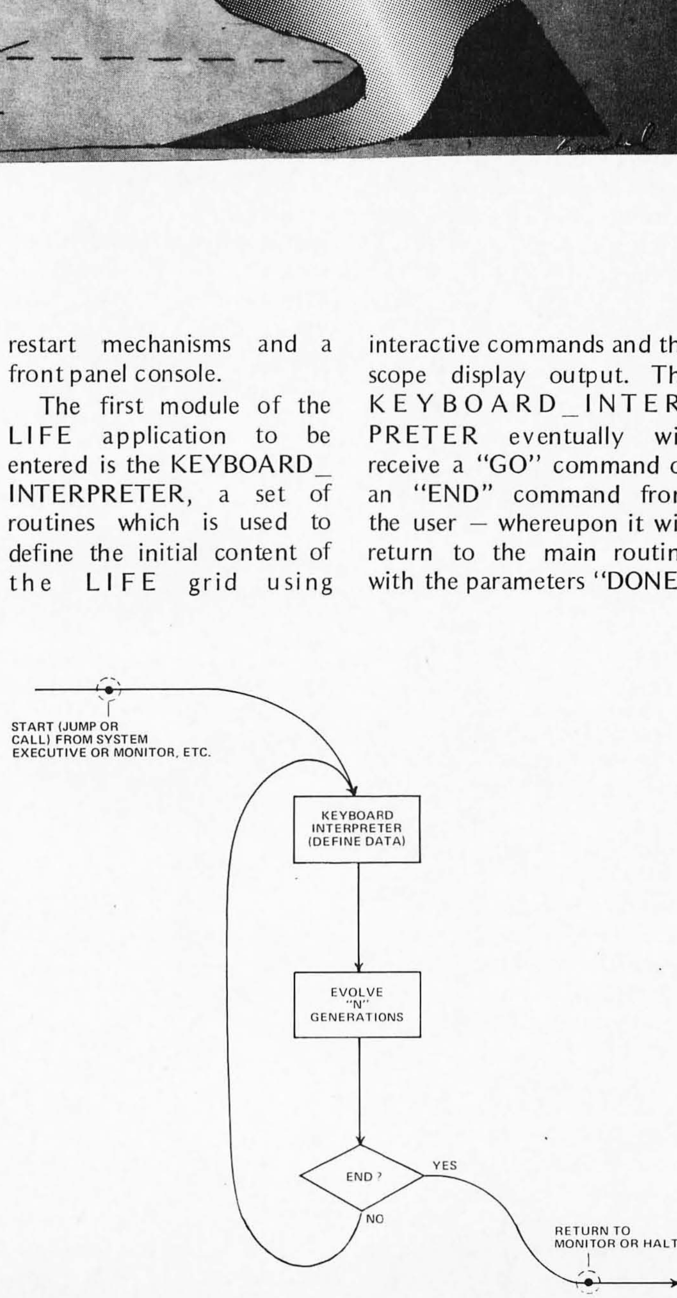


Fig. 2. LIFE program flow of control.

Fig. 3. The main control routine of LIFE specified in a procedure-oriented language . . .

```
1  LIFE:
2      PROGRAM;
3          DONE = FALSE;
4          DO UNTIL DONE = TRUE;
5              CALL KEYBOARD_INTERPRETER (N,DONE);
6              DO FOR I = 1 TO N;
7                  CALL GENERATION;
8              END;
9          END;
10         RETURN; /* TO EXECUTIVE, MONITOR, OR JUST HALT */
11         CLOSE LIFE;
```

Subroutines Referenced by LIFE:

KEYBOARD_INTERPRETER . . . This is the routine which looks at the interactive keyboard and interprets user actions such as specifying initial patterns, modifying patterns, etc. N is defined by the GO command which causes return from this subroutine to LIFE.

GENERATION . . . This is the routine which is used to evolve one generation of the life matrix and display the result. Since the entire matrix is kept in software by *GENERATION* until after a new matrix has been evolved, there will never be any partially updated patterns on the scope.

Data (8 bit bytes) used by LIFE at this level:

FALSE – the value "0".

TRUE – the value "1".

DONE – variable set by *KEYBOARD_INTERPRETER* after a user command (GO) to start execution.

N – a variable set by user interaction in *KEYBOARD_INTERPRETER* giving the number of generations to evolve.

I – a temporary loop index variable.

. . . the problem is how to preserve the blissful innocence of simplicity in the face of the wordly forces of complexity.

Fig. 3 uses a "procedure-oriented language" (see the box accompanying this article) to specify the program in a more explicit and compact form than is possible with a flow chart. Each line of the program as specified in Fig. 3 could potentially be compiled by an appropriate compiler – but for the purposes of most home computer systems, generation of code from this model would be done by hand. The outer loop is performed by a "DO UNTIL" construct starting at line 4 and extending through line 9. The program elements found on lines 5 to 8 are executed over and over again until DONE is found to be equal to logical 1 or "TRUE" when a test is made at the END statement of line 9. A "DO FOR" loop is used to sequence "N" calls to a subroutine called GENERATION which does the actual work of computing the next generation content and then displaying it on the scope. The remainder of Fig. 3 summarizes the data and subroutines referenced by LIFE.

From this point, the LIFE Line can extend in two directions. In order to have a complete LIFE program, both areas have to be traversed – the KEYBOARD_INTERPRETER and the GENERATION routine . . .but the partitioning has nicely separated the two problems. The simpler and most self-contained of the two segments is the GENERATION algorithm, so I'll turn attention to it next.

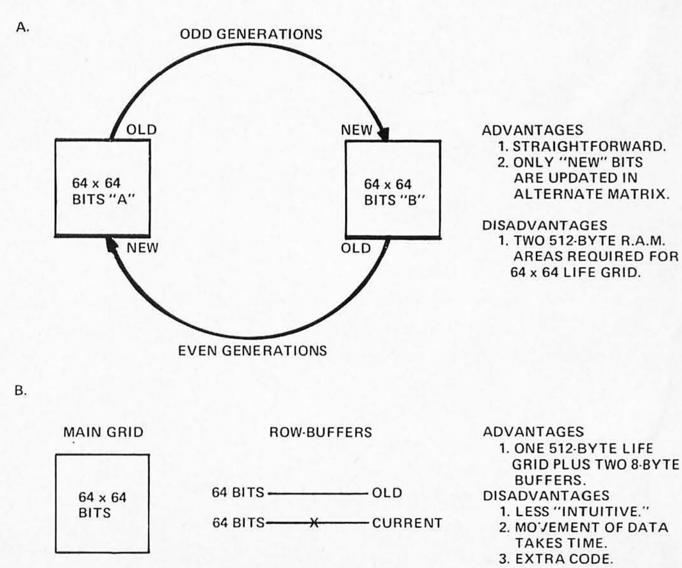
Grid Scanning Strategies

The GENERATION subroutines of the LIFE program has as its design goal the transformation of one complete LIFE grid pattern into the "next" complete pattern. The rules of the Game of LIFE – the "facts of life" – must be applied to each location in the grid to compute the next value of that location. Fig. 4 illustrates two potential strategies for computing the next generation – methods of scanning the grid to compute one location at a time.

The first strategy, Fig. 4(a), is to employ alternate copies of a complete LIFE grid of 64 by 64 points. If generations are numbered consecutively, the generation algorithm would transform copy A into a "next" copy in B on odd generations, and complete the cycle by transforming copy B into a "next" copy in the A grid on even generations. Since each grid requires 4096 bits – which can be packed into 512 bytes – a total of 1024 bytes is required for data storage if this method is used. The primary advantages of this method are its "straightforward" nature and its separation of old and new data at all times.

A second strategy is illustrated in Fig. 4(b), the strategy of using alternate row-buffers with only one

Fig. 4. The LIFE evolution algorithm – matrix scanning techniques which preserve relevant old information while creating new information in overlapping storage areas.



main grid copy. Two 64 bit rows must be maintained — the last previous row and the current row — as 8 byte copies. These copies contain information *prior to updating* in the row by row scan down the matrix. The main advantage is a saving of data areas (partially offset by more complicated software). The main disadvantages are its less "intuitive" nature, the extra time required to make data copies, and a slightly larger program.

The choice between these two methods is primarily one of the amount of storage to be devoted to data. The tradeoff is in favor of the double matrix method when very small LIFE matrix sizes are considered. The extra 8 bytes required for a second copy of an 8 by 8 grid of bits hardly compares to the programming cost of the alternate row-buffer strategy. When large matrices are considered, however, the memory requirements of an extra copy of the data are considerable, but the programming involved is no more difficult. For example, consider the limit of an 8 bit indexing method — a 256 by 256 grid. This will require a total of 8192 bytes for *each copy* of the LIFE grid. Two copies of the LIFE grid would use up 16k bytes, or one fourth of the addressing space of a typical contemporary microcomputer, and all of the addressing space of an 8008 microcomputer! At the 64 x 64 bit level, the tradeoff is much closer to the break-even point, but I expect to find at least 100 bytes saved as a result of using the row-buffer method. An assumption which is also being made when the alternate row-buffer method is used is that the scope display or TV display you use for output will have its own refresh memory so that the "old" pattern can be held during computation of the new. If this is not the case, a less desirable output in which partially updated patterns are seen will be the result. (Counting the CRT refresh, the method of Fig. 4(a) thus requires three full copies of the matrix information, and the method of Fig. 4(b) requires two full copies.)

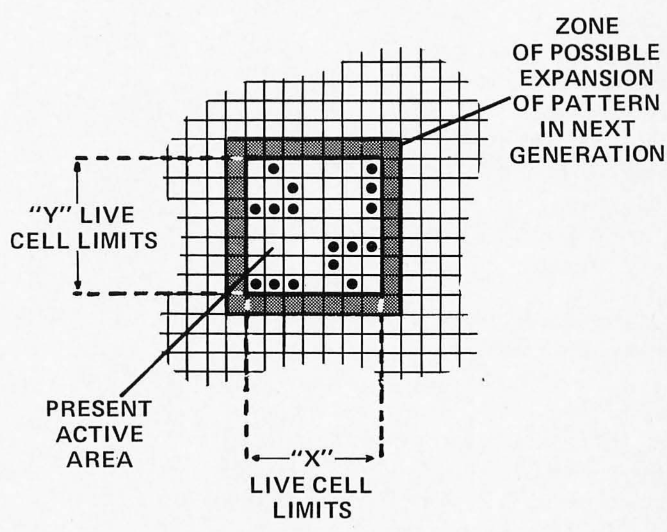
An objective: Split the processing into moderately large segments, each of which is less complicated than the program taken as a whole.

Active Area Optimization

With the choice of a matrix scanning strategy — the alternate row-buffer method — another consideration in designing the generation algorithm is a computation time optimization method. There is no real need to calculate a new value of every cell in a mostly empty LIFE grid. If I only have one glider with its corner at location (34, 27) of the grid, why should I compute any new generation information outside the area which could possibly be affected by the present pattern's evolution? Again, the savings in computation time using active area optimization depend upon the size of the grid. If most patterns occupy the full grid, then very little will be saved — for the small 8 x 8 grid "straw man" used in discussing scanning strategy, there would also be no point to active area optimization. But with a huge 256 by 256 grid, and an 8 by 8 active area, this optimization might mean the difference between a 15 minute computation and a 1 or 2 second computation of the next generation.

Fig. 5 illustrates the concept of active area optimization in a LIFE program. The current generation's active area is defined as the set of X and Y limits on the extent of live cells in the grid. In Fig. 5, the active area is the inner square of 7 x 7 = 49 grid locations. In computing the next generation, a box which is one location wider in each of the four cardinal directions is the "zone of possible expansion" for the pattern. In Fig. 5, this zone is the outer box of 9 by 9 locations. The computation of "next generation" values need only be carried out for the 81 grid locations bounded by the outer limits of the zone of possible expansion. Thus in the case of the 64 by 64 matrix of LIFE points, this optimization for the pattern of Fig. 5 will limit the program to calculation of 81 new points versus the 4096 points which would be calculated if at least one cell was found at each of the minimum (0) and maximum (63) values of the X and Y

Fig. 5. Active area optimization — never compute more than the absolute minimum if speed is at a premium.



The "facts of life" must be applied to each location in the grid to compute the next value – cell or no cell – of that location.

coordinates. This case yields a savings of 98% of the maximum generation to generation computing time.

The GENERATION Subroutine

Fig. 6 illustrates the code of the GENERATION routine, specified in a procedure-oriented language, along with notes on further subroutines and data requirements. The procedure starts by initializing the data used for the scan of the matrix, in lines 3 through 15. THIS and THAT are used to alternately reference the 0 and 1 copies of an 8 byte data item called a 2 by 8 byte data area called "TEMP". (Subscripts, like in XPL and PL/M are taken to run 0 through the dimension minus 1.) NRMIN, NRMAX, NCMIN, and NCMAX are used to keep track of the new active area limits after this generation is computed; NROWMIN, NROWMAX, NCOLMIN and NCOLMAX are originally initialized by the KEYBOARD_INTERPRETER and are updated by LIMITCHECK after each generation is calculated – using the new active area limits.

The actual scan of the grid of LIFE, stored in the data area called LIFEBITS, is

Fig. 6. The GENERATION routine specified in a procedure-oriented language . . .

```
 1 GENERATION:
 2   PROCEDURE;
 3   THIS = 0;  /* INITIALIZE POINTERS TO TEMPORARY ROW  */
 4   THAT = 1;  /*     COPY VARIABLE "TEMP"              */
 5   DO FOR I = 0 TO 7;
 6      IF NROWMIN = 0 THEN
 7         TEMP (THAT,I) = LIFEBITS(63,I);
 8      ELSE
 9         TEMP (THAT,I) = 0;
10      /* THIS ESTABLISHED WRAP-AROUND BOUNDARY CONDITION */
11   END;
12   NRMIN = 99;  /* THEN INITIALIZE ACTIVE AREA LIMITS  */
13   NRMAX = 0;
14   NCMIN = 99;
15   NCMAX = 0;
16 ROW_LOOP:
17   DO FOR IROW = NROWMIN TO NROWMAX; /* SCAN ACTIVE ROWS ONLY */
18      DO FOR I = 0 TO 7;  /* COPY THIS ROW TO TEMPORARY */
19         TEMP (THIS,I) = LIFEBITS (IROW,I);
20      END;
21      DO FOR ICOL = NCOLMIN TO NCOLMAX; /* SCAN ACTIVE COLUMNS ONLY */
22         CALL FACTS_OF_LIFE (IROW, ICOL);
23      END;
24      X = THIS;
25      THIS = THAT;
26      THAT = X; /* THIS SWITCHES BUFFERS */
27   END;
28   CALL LIMITCHECK;
29   CALL DISPLAY;
30   CLOSE GENERATION;
```

Subroutines Referenced by GENERATION:

EVOLVER . . . This is the routine used to calculate the next value of the ICOLth bit in the IROWth row of LIFEBITS using the current value of the next row, the saved value in TEMP of the previous row, and the saved value in TEMP of the current row before updating.

LIMITCHECK . . . This is the routine used to calculate the next values of NROWMIN, NROWMAX, NCOLMIN, NCOLMAX using the current values of NRMIN, NRMAX, NCMIN and NCMAX.

DISPLAY . . . This routine transfers the LIFEBITS data to the display, on whatever kind of device you have.

Data (8 bit bytes) used by GENERATION at this level:

X = TEMPORARY
I = temporary index (not the same as the I in Fig. 3)
ICOL = index for column scanning . . .
IROW = index for row scanning . . .
NCMAX = current maximum column index of live cells
NCMIN = current minimum column index of live cells
NRMAX = current maximum row index of live cells
NRMIN = current minimum row index of live cells

Data (8 bit bytes) used by GENERATION but shared with the whole program:

THIS = current line copy index into TEMP.
THAT = previous line copy index into TEMP.
TEMP = 2 by 8 array of bytes containing 2 64-bit rows.
NROWMIN = minimum row index of live cells.
NROWMAX = maximum row index of live cells.
NCOLMIN = minimum column index of live cells.
NCOLMAX = maximum column index of live cells.
LIFEBITS = 64 by 8 array of bytes containing 64 rows of 64 bits.

Assumptions:

LIFEBITS, NROWMIN, NROWMAX, NCOLMIN, NCOLMAX are initialized in *KEYBOARD_ INTERPRETER* for the first time prior to entry – and retain old values across multiple executions of *GENERATION* thereafter.

Fig. 7. The LIMITCHECK routine specified in a procedure-oriented language . . .

```
LIMITCHECK:
    PROCEDURE;
    /* CALCULATE NEXT ROW LIMITS */
    IF NRMIN-1 < NROWMIN THEN NROWMIN = NRMIN-1;
    IF NRMAX+1 > NROWMAX THEN NROWMAX = NRMAX+1;
    IF NROWMAX > 63 THEN NROWMAX = 63;
    IF NROWMIN < 0 THEN NROWMIN = 0;
    /* CALCULATE NEXT COLUMN LIMITS */
    IF NCMIN-1 < NCOLMIN THEN NCOLMIN = NCMIN-1;
    IF NCMAX+1 > NCOLMAX THEN NCOLMAX = NCMAX+1;
    IF NCOLMAX > 63 THEN NCOLMAX = 63;
    IF NCOLMIN < 0 THEN NCOLMIN = 0;
    CLOSE LIMITCHECK:
```

Subroutines Referenced by LIMITCHECK:

None.

Data (8 bit bytes) used by LIMITCHECK but shared with the whole program:

NCOLMAX, NCOLMIN, NROWMAX, NROWMIN
NRMAX, NRMIN, NCMAX, NCMIN *(see Fig. 6)*

Assumptions:

The arithmetic of the comparisons in this routine is done using signed two's complement arithmetic – thus a negative number results if 0 - 1 is calculated . . . this is consistent with code generation on most 8 bit micros.

performed by the set of DO groups beginning with ROW_LOOP at line 16. For each row of the matrix, ROW_LOOP first copies the row into TEMP as the THIS copy (the THAT copy is left over from initialization the first time at lines 5 to 11, or from the previous ROW_LOOP iteration thereafter). Following the copying operation, another DO FOR loop goes from NCOLMIN to NCOLMAX applying the FACTS_OF_LIFE to each grid position in the current (THIS) row as saved in TEMP. New data is stored back into LIFEBITS by FACTS_OF_LIFE. At the end of the row loop, prior to reiteration, the THIS and THAT copies of temp are switched by changing the indices. What was THIS row becomes THAT row with respect to the next row to be computed.

After all the rows have been computed, line 28 is reached. Line 28 calls subroutine LIMITCHECK to compute the next generation's active area computation limits using the results of this generation. Line 29 then calls a module named DISPLAY to copy the results of GENERATION into the output display device. The LIMITCHECK routine simply performs comparisons and updating – Fig. 7 illustrates the high level language description of its logic.

Computing The Facts of LIFE . . .

Fig. 8 contains the information on implementing the Facts of LIFE in a programmed set of instructions. The computation is divided into two major parts. The first part is to determine the STATE of the bit being updated, where "STATE" is a number from 0 to 8 as described in LIFE Line 1 last month. The second major step is to evolve the grid location using its current value and the STATE.

FACTS_OF_LIFE begins by performing left and bottom boundary wrap-around checks by adjusting indices. Lines 8 to 18 determine the current STATE by referencing all 8 grid locations surrounding the location being computed at (IROW, ICOL). In determining the state, the subroutines TGET and LGET

Two copies of a 256 by 256 grid would require more memory than (for example) an 8008 can address if you want to have programs along with your data.

Why should I compute any new generation information outside the area which could possibly be affected by the present pattern's evolution?

Fig. 8. The FACTS_OF_LIFE routine specified in a procedure-oriented language. FACTS_OF_LIFE does the actual calculation of the next value for the LIFEBITS location at the IROW^th row and ICOL^th column based upon the previous value of the 8 neighboring locations. (The state defined in LIFE Line 1, last month.) This routine implements the rules described in BYTE #1, page 73.

```
1  FACTS_OF_LIFE:
2     PROCEDURE (IROW,ICOL);
3     M = IROW + 1;
4     IF M >63 THEN M = 0; /* BOTTOM BOUNDARY WRAP CONDITION */
5     N = ICOL - 1;
6     IF N < 0 THEN N = 63; /* LEFT BOUNDARY WRAP CONDITION */
7  DETERMINE_STATE:
8     STATE = TGET (THAT,N);
9     STATE = STATE + TGET (THIS,N);
10    STATE = STATE + LGET (M,N);
11    N = ICOL;
12    STATE = STATE + TGET (THAT,N);
13    STATE = STATE + LGET (M,N);
14    N = ICOL + 1;
15    IF N > 63 THEN N = 0; /* RIGHT BOUNDARY WRAP CONDITION */
16    STATE = STATE + TGET (THAT,N);
17    STATE = STATE + TGET (THIS,N);
18    STATE = STATE + LGET (M,N);
19 EVOLVEIT:
20    NEWCELL = 0; /* DEFAULT EMPTY LOCATION UNLESS OTHERWISE */
21    OLDCELL = TGET (THIS, ICOL);
22    IF OLDCELL = 1 THEN DO;
23       IF STATE = 2 OR STATE = 3 THEN NEWCELL = 1;
24    END;
25    ELSE DO;
26       IF STATE = 3 THEN NEWCELL = 1;
27    END;
28    CALL LPUT (IROW, ICOL, NEWCELL);
29    IF NEWCELL = 1 THEN CALL SETLIMIT (IROW, ICOL);
30    CLOSE   FACTS_OF_LIFE;
```

Subroutines Referenced by FACTS_OF_LIFE:

TGET. . . This is a "function" subroutine which returns an 8 bit value (for example in an accumulator when you generate code) of 00000001 or 00000000 depending upon whether or not a referenced column in one of the two temporary line copies in TEMP is 1 or 0 respectively. The first argument tells which line of the two, and the second argument tells which column (0 to 63) is to be retrieved.

LGET. . . This is a "function" subroutine which returns an 8 bit value similar to TGET, but taken instead from the bit value at a specified row/column location of LIFEBITS.

LPUT. . . This subroutine is used to set a new value into the specified row/column location of LIFEBITS.

NOTE: The routines LGET and LPUT will be referenced from the KEYBOARD_INTERPRETER *routine in the course of manipulating data when setting up a life pattern.*

SETLIMIT. . . This subroutine is used to check the current active region limits when the result of the facts of life indicate a live cell.

Data (8 bit bytes) used by FACTS_OF_LIFE at this level:

IROW = Parameter passed from *GENERATION.*
ICOL = Parameter passed from *GENERATION.*
M = temporary, row index.
N = temporary, column index.
STATE = count of "on" bits in neighborhood of *IROW, ICOL.*
OLDCELL = temporary copy of old cell at *IROW, ICOL.*
NEWCELL = new value for location *IROW, ICOL.*

Data (8 bit bytes) used by FACTS_OF_LIFE but shared with the whole program:

THAT, THIS (see Fig. 6)

What was THIS row becomes THAT row with respect to the next row to be computed. (What's in a name? A pointer of course!)

are used to reference bits in TEMP and LIFEBITS respectively, using appropriate bit location indices. The values returned by these two "function subroutines" are either 0 or 1 in all cases – thus counting the number of "on" cells consists of adding up all the TGET or LGET references required to examine neighboring grid locations.

Once the STATE of the grid location is determined, the Facts of LIFE are implemented by examining the *positive* cases of an "on" (live cell) value for the grid location. A cell will be in the grid location for the next generation in only two cases: If the old content of the location was a live cell and the STATE is 2 or 3; or if the old content of the location is 0 (no cell) and the STATE is 3. A default of NEWCELL = 0 covers all the other cases if these two do not hold. Line 28 stashes the new value away in LIFEBITS with subroutine LPUT, and if the new value of the grid location

Fig. 9. The SETLIMIT routine specified in a procedure-oriented language.

```
SETLIMIT:
    PROCEDURE (IROW,ICOL);
    IF IROW < NRMIN THEN NRMIN = IROW;
    IF IROW > NRMAX THEN NRMAX = IROW;
    IF ICOL < NCMIN THEN NCMIN = ICOL;
    IF ICOL > NCMAX THEN NCMAX = ICOL;
    CLOSE SETLIMIT;
```

Subroutines Referenced by SETLIMIT:
None.

Data (8 bit bytes) used by SETLIMIT at this level:

IROW = parameter passed from *FACTS_OF_LIFE.*
ICOL = parameter passed from *FACTS_OF_LIFE.*

Data (8 bit bytes) used by SETLIMIT but shared with the whole program:

NRMIN, NRMAX, NCMIN, NCMAX *(see Fig. 6)*

is a live cell, SETLIMIT is called (see Fig. 9) in order to update the active area pointers NRMAX, NRMIN, NCMAX and NCMIN.

Where Does the LIFE Application Stand?

An alternative to the pyramid structure way of viewing programming program designs introduced at the beginning of this article is a "tree" notation showing the heirarchy of modules in the application. The "Tree of LIFE" is shown in Fig. 10 as it exists in materials printed to date. The next installment of LIFE Line will explore the left hand branch of the tree diagram by a similar presentation of a KEYBOARD_INTERPRETER algorithm.

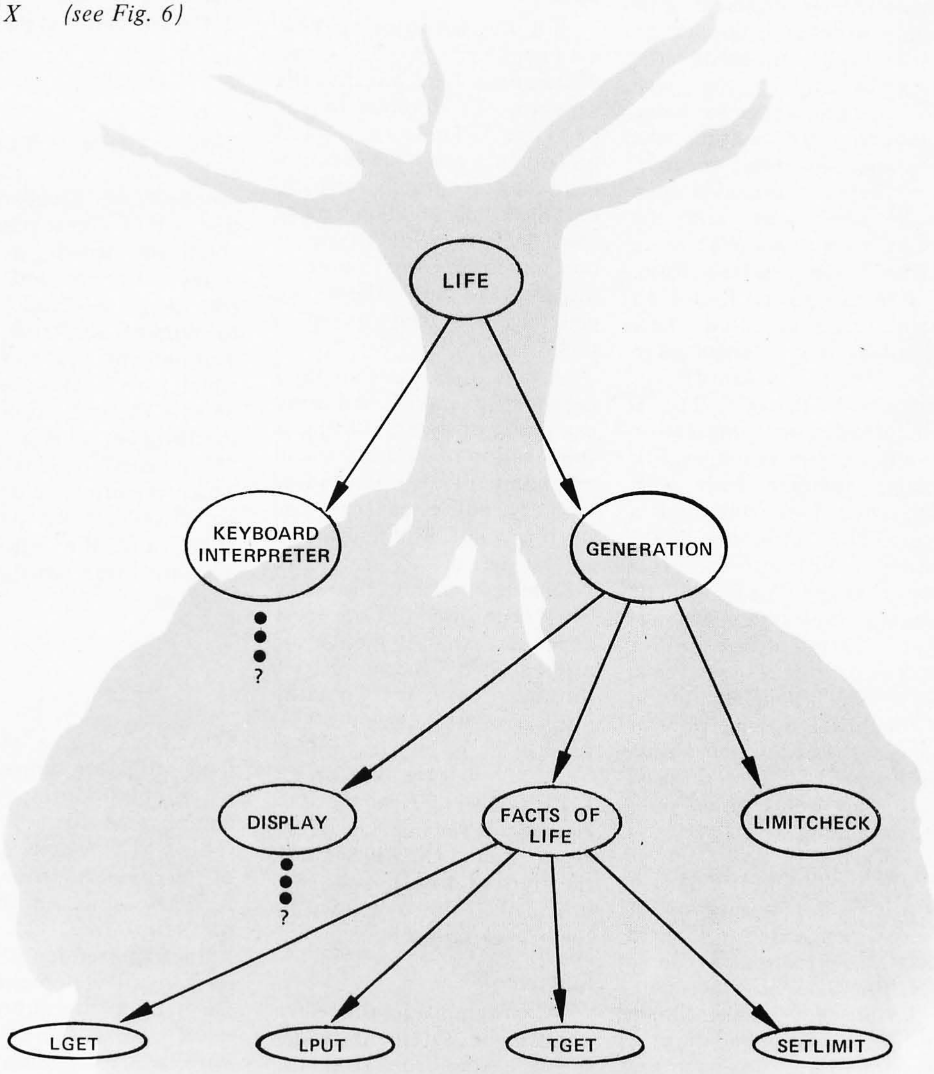


Fig. 10. The Tree of LIFE.

LIFE Line 2 Addendum

Procedure-Oriented Computer Languages

The examples of programs accompanying two articles in this issue have been constructed in a procedure-oriented language. This method of program representation is compact and complete. In principle, one could write a compiler to automatically translate the programs written this way into machine codes for some computer. By writing the programs in this manner, more detail is provided than in a flow chart, and the program is retained in a machine independent form.

The particular representation used here resembles several languages in the "PL/1" family of computer languages, but is not intended for compilation by any existing compiler. For readers familiar with such languages, you will find a strong PL/1 influence and a moderate XPL influence. In a future issue BYTE will be running articles on a language specifically designed for microcomputer systems, PL/M, which is an adaptation of the XPL language for 8-bit machines. For the time being, this representation is used with some notes to aid your understanding.

Programs and Procedures

A *program* is a group of lines which extends from a PROGRAM statement to a matching CLOSE statement. It is intended as the "main routine" of an application. A *procedure* is a similar group of lines which extends from a PROCEDURE statement to its CLOSE statement. A procedure may have *parameters* indicated in the PROCEDURE statement, and may be *called* as a "subroutine" from a program or another procedure. A procedure may be called in a "function" sense as well, in which case a RETURN statement would be required to set a value.

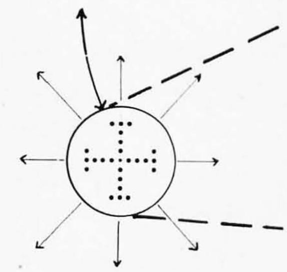
Data

For the purposes of these examples, no "data declarations" are put into the programs to complicate the picture. Instead, each example has a section following it which verbally describes each data name used. Only one "data type" is considered at this point – integers – and these are generally assumed to be 8 bits.

Arrays of integers are used in several examples. An array is a group of bytes, starting at the location of its *address* and extending through ascending memory addresses from the starting point. The purpose of an array is to reference "elements" within the array by "subscripts". For these examples, the elements are referenced by the numbers 0 through "n-1" for an array dimension of length "n". If LIFEBITS is an array of 64 by 8 bytes, then LIFEBITS(63,7) is the last element of the last row of the array, and LIFEBITS(I,J) is the byte at row I, column J provided I and J are within the proper ranges.

Statements

A program or procedure consists of statements which specify what the computer should do. In a machine language, these would correspond to the selected operation codes of the computer which is being programmed. For a high level language, one statement typically represents several machine instructions. In these the high level language statement has a "semantic intent" – a definition of its operation – which can be translated into the lower level machine language. In these examples several types of statements are employed . . .



"IF . . . THEN . . . ELSE . . ." constructs are used for notation of decisions. The first set of ellipses indicate a condition which is to be tested. The second set of ellipses in the model is used to stand for the "true part" – a statement (or DO group) which is to be executed if and only if the condition is true; the third set of ellipses is the "false part" – a statement which is only executed if the condition is false. The word ELSE and the whole false part are often omitted if not needed.

"CALL X" is a statement used to call a subroutine, in its simplest form. A more complicated form is to say CALL X(Z) where Z is a set of "arguments" to be passed to the routine. Another form of subroutine call is the "function reference" in an assignment statement, where the name of the subroutine is used as a term in an arithmetic expression.

"assignment" – a statement of the form "X = Y;" is called the *assignment statement.* Y is "evaluated" and the result is moved into X when the statement is executed. If X or Y have subscripts as in "TEMP(THIS,I) = LIFEBITS(IROW,I);" then the subscripts (such as "THIS,I" and "IROW,I" in the example) are used to reference the name as an array and pick particular bytes.

"DO groups" – a grouping of several statements beginning with a "DO" statement and running through a corresponding "END" is used to collect statements for a logical purpose. In "DO FOR I = 0 to 7;" this purpose is to execute the next few statements through the corresponding "END;" 8 times with I ranging from 0 to 7. "DO UNTIL DONE=TRUE;" is an example of a group which is repeated indefinitely until a condition is met at the END. "DO FOREVER" is a handy way of noting a group to be repeated over and over with no end test, a practice often frowned upon.

A QUICK Test of Keyboards

This indicator circuit can be used to advantage when analyzing keyboards using techniques described in BYTE #1, "Deciphering the Mystery Keyboard," page 62.

After completing the assembly of a keyboard late one night, I wanted to check the keyboard out for proper operation. So I picked up my VOM and started looking at the voltage levels on the output pins of the keyboard, since I do not have a CRT terminal or any other ASCII device available. Well, being a software type, I kinda felt a little frustrated since I am generally used to being able to see all the bits of a bit pattern at the same time. The solution was very simple, inexpensive, and quickly allowed the bit pattern on the keyboard output pins to be viewed as a bit pattern. Fig. 1 shows the system used. The LEDs are lit or unlit depending on the key pressed and held. The pattern produced by the LEDs will display the bits of the character generated by the key pressed on the keyboard. Keyboards which generate ASCII, EBIDIC, or whatever could be checked out quickly with this system.

Example

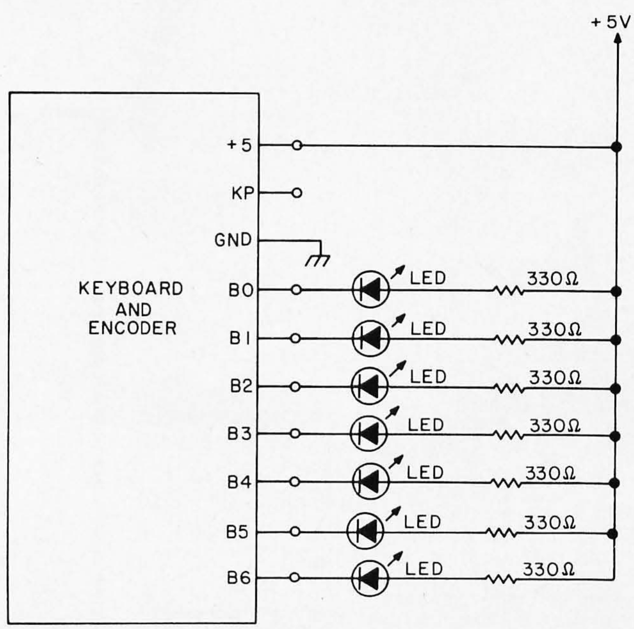
A keyboard which generates ASCII coded characters has the "A" key pressed and held. The LED bit pattern would look like this:

0-LED on, logic level high
●-LED off, logic level low

0 ●●● ●●0 ASCII character
code for "A"
bit Ø 123 456
1 000 001

It should be pointed out that this test method will work without modification with diode encoded keyboards such as Southwest Technical Products KBD-2 keyboard (which is the keyboard I assembled and tested with the above method). However, some keyboards may generate an inverted code which shouldn't be a problem. Some keyboards (surplus and perhaps new) with more sophisticated debouncing techniques may not work with this test method without some additional components. For example, some keyboards have a bus-oriented tri-state MOS output without sufficient drive to light the diode lamps; you would need a buffer gate in this case, as well as an output data strobe. Other keyboards require an active "read" operation in which a pulse is supplied to reset flip flops acknowledging CPU acceptance of data.

Fig. 1. Examining Keyboard Outputs with LED Indicators. A TTL-compatible output can drive the typical LED with about 10 milliamperes in an active low state.



by
Don R. Walters
3505 Edgewood Dr.
Ann Arbor MI 48104

GODBOUT

BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

FLYER HYPE---WE SELL CAPACITORS, CLOCKS, RESISTORS, LINEARS, TTL, CMOS, MOS, KEYBOARDS, KITS, MUSICIAN'S KITS, SOCKETS, FETS, TRANSISTORS, TOOLS, HOBBYIST'S SPECIALS, READOUTS, MORE KITS, AND SPECIAL ITEMS. SEND A STAMP AND YOUR ADDRESS TO GET OUR NEW FLYER LISTING ALL OUR STUFF.

VECTOR WIRE PENCIL



$9.50

+1 POUND SHIPPING

VECTOR P173 WIRING PENCIL---A HAND HELD, UNDER 1 OZ. TOOL USED TO HELP GUIDE AND WRAP #36 INSULATED WIRE (FROM A SELF-CONTAINED BOBBIN) ONTO COMPONENT TERMINALS. MAKE CONNECTIONS BY SOLDERING; HEAT FROM THE IRON MELTS THE INSULATION, PERMITTING A GOOD SOLDER BOND. THIS IS A FAST WAY TO WORK WITH COMPONENTS AND VECTORBOARD. COMES WITH TOOL, AN INSTALLED BOBBIN OF 250 FT. GREEN WIRE, AND AN EXTRA BOBBIN OF RED WIRE AS WELL AS INSTRUCTIONS..................$9.50

---AND TO GO WITH YOUR WIRE PENCIL---

VECTORBOARD---TYPE 169P84-062, PRE-PUNCHED VECTORBOARD WITH HOLES ON .1" CENTERS (PERFECT FOR ICs). 8½" X 17", 1/16" THICK EPOXY GLASS................$8.95 + 1 lb. SPHG/BOARD.

1/2 AMP power supply kits

---LESS CASE AND HARDWARE---

REGULATED, SHORT-PROOF, THERMALLY LIMITED. HALF AN AMP CONTINUOUS OR 1 AMP INTERMITTENT. SAVE YOURSELF THE HEADACHE OF DESIGNING YOUR OWN; USE ONE OF OURS. SPECIFY 5, 6, 8, OR 12 VOLTS. $7.95 + 2 LBS.

ICs

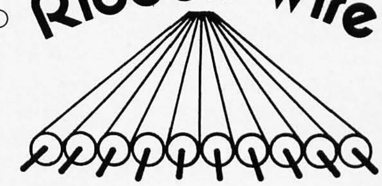
5203 2K EROM
---$14.50---

5204 4K EROM
---$24.50---

8008 8 BIT MICROPROCESSOR
---$27.95---

8 BIT CHIP SET - 8008 + 8-2102s
---$42.50---

Ribbon Wire



This is a flat, multiconductor wire, available in multiples of 10 conductors up to 100 conductors. Comes in 20 ft. lengths only. Cost is 1¢ per conductor foot; for example, a 20 conductor cable 20 ft. long = $4.00

See facing page for computer news

SOCKET SPECIAL

SOCKET TO YOU! WE MADE A SPECIAL PURCHASE OF PC MOUNT SOLDERTAIL SOCKETS, INCLUDING DISCONTINUED TYPES AND A LOT OF 14 PIN SOCKETS THAT REQUIRE HIGHER THAN SPEC INSERTION FORCE. BRAND NEW FROM A LEADING MFGR OF TOP QUALITY IC SOCKETS.

(G) = GOLD PLATED CONTACTS (T) = BRIGHT TIN PLATE

T 14 PIN	11/$1.95	T 40 PIN	5/$2.95
G 14 PIN	10/$1.95	LOW PROFILE SOCKETS:	
T 24 PIN	5/$1.95	G 16 PIN	10/$1.95
T 28 PIN	5/$1.95	G 40 PIN	5/$2.95

CAVE grafix™

Hobbywrap tool

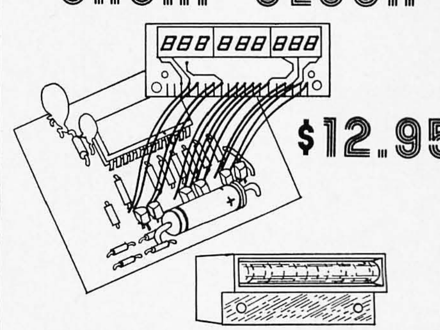


$41.95

HOBBYWRAP TOOL---WHY SOLDER YOUR PROTOTYPES? WIRE WRAPPED CONNECTIONS PROTECT COMPONENTS FROM HEAT, ARE EASILY MODIFIED FOR CORRECTIONS OR CHANGES, MAKE CONNECTIONS THAT ARE BETTER THAN SOLDER, AND CAN SPEED UP THE TIME REQUIRED FOR ASSEMBLY OF COMPLEX DIGITAL PROJECTS. IF THE HIGH COST OF WRAPPING PUTS YOU OFF, LOOK AT OURS..YOU GET THE TOOL (RECHARGEABLE SO YOU DON'T HAVE TO DEAL WITH TRAILING CORDS IN TIGHT PLACES), BIT, CHARGER, NICADS, AND INSTRUCTIONS.

TERMS: ADD 50¢ TO ORDERS UNDER $10. ADD SHIPPING AS SHOWN. CAL RES ADD TAX. NO COD.

CHEAP CLOCK



$12.95

This kit has gone over very well. Comes less case and transformer, but includes everything else. Parts mount on motherboard; readout + lens mount on daughterboard, and the readout may be remoted. The readout-board-lens combination alone has sold for $14.95 nationally. POSTPAID

Modular Construction, OR Why Not Do It Yourself ?

Ask a computer person what "hardware" is and you'll often get the answer "gates, MSI, ROMs and microcomputers". Ask an astronautical engineer what "hardware" is and you'll often get the answer "boosters, fuel tanks and space shuttles . . ." Here in this article, Don Walters covers a few of the "nuts and bolts" of packaging hardware for electronics. Which concept of hardware you wish to use depends upon the point of view – your focus of attention.

. . .CARL

Modular construction is nothing new. In fact, the concept has been around for a very long time. Many companies which make computers or computer systems use this method of construction for their products. For example, a company which makes an industrial process control computer system might buy the power supply, printed circuit board card cage, physical system enclosure, CPU or memory, or peripherals from one or more other companies. The company which makes the industrial process control computer system then assembles the parts together, writes the necessary software, and adds whatever other special hardware is needed. The end product is something new, but is still made up out of all the subassemblies (modules). But more importantly, because the system is made up of modules, if a module becomes defective it can be removed, repaired or replaced without having to do a lot of unwiring and rewiring or performing major surgery on the system.

Well, you might think, that is fine for companies and people who can afford to buy the modules from the various manufacturers. But I can't afford that, so my system will have to be put together as best I can. Well, guess what? You too can put your system together in a modular form (assuming you haven't purchased a kit which tends to limit your ability to go the modular route, or already has done it for you, such as is the case with the many existing inexpensive computer systems).

Figs. 1, 2, 3 and 4 show how easy it is to fabricate printed circuit board card cages, cabinets, printed circuit boards, and even modular power supplies. True, it always looks good on paper, but if you think out the project and are a little careful and persistent, you should be able to fabricate the modules of your computer system at a fraction of the cost that commercial units would cost, plus your system will have a good

Fig. 1. Printed circuit card cage concept. Card cages can be fabricated along these lines. A short cut would be to strip and modify a cage obtained from a surplus dealer.

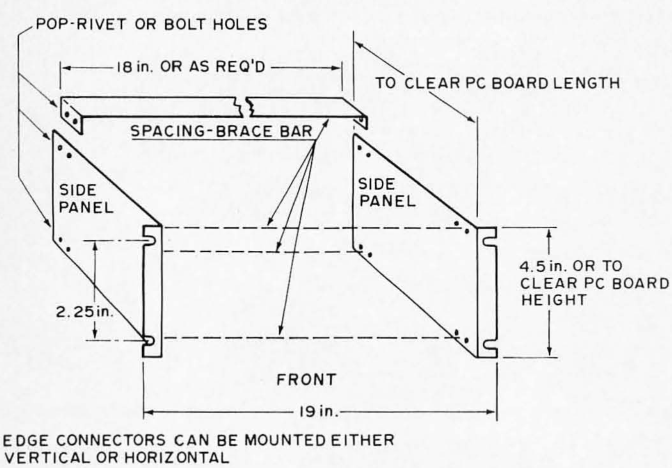
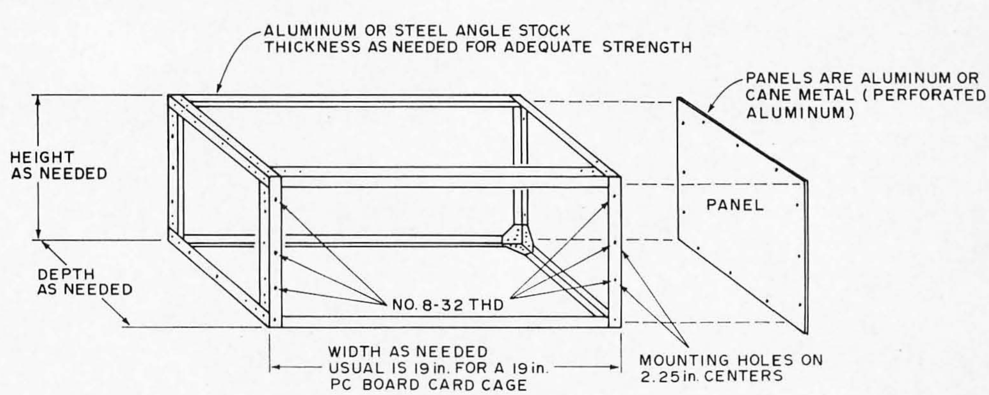
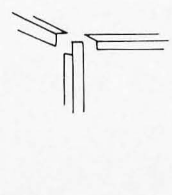
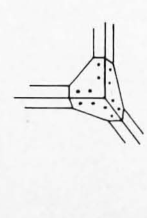


Fig. 2. Cabinet Fabrication. Imagination, metal plates and angle stock are blended into a computer cabinet for your system.



by
Don R. Walters
3505 Edgewood Dr.
Ann Arbor MI 48104





professional-looking appearance.

The printed circuit board card cage of Fig. 1 is simple to fabricate from locally available materials (hobby shops, hardware stores, and perhaps lumber yards are a good source of materials).

The side panels of the PC board card cage can be made from heavy gauge aluminum. Aluminum door kick panels available in most hardware stores are good sources for the aluminum of the side panels. The spacing-brace bar is, again, something found in most hardware stores in the form of long thin (approximately 1/16 inch thick) aluminum bars. The card cage assembly can be fastened together with pop rivets or nuts and bolts (6-32 binder head machine screws are about the right size). Card guides can be made from channeled plastic strips, aluminum or even wood. The card guides can be fastened in the card cage with glue or drilled and bolted in. PC board edge connectors can be mounted on the back edge of the back spacing-brace bars in a vertical position by drilling and tapping holes in the back edge of the back spacing-brace bars. Edge connectors could also be mounted horizontally on a piece of aluminum and then fastened to the back spacing-brace bars.

The cabinet of Fig. 2 is another item that can be fabricated easily. The cabinet can be made from aluminum or steel angle stock. For small cabinets, aluminum is probably more desirable since it is lighter. However, steel is more easily welded and would be better for larger cabinets or relay rack type enclosures.

Printed circuit boards are not that hard to make. Really! The big secret to success is to thoroughly clean the copper surface of the PC board. The best way to do this is to use an abrasive cleanser such as COMET, AJAX, etc., and a little elbow grease. Then use the method[1] that best suits your needs for laying out the foil pattern and etching the board. PC board edge connector foil pads are not too hard to put on a board, even if you use an etch resist pen to do it. There are also a couple of companies who sell a rubber stamp for edge connector foil pads, IC pin pads, etc. Etching PC boards is not difficult either. Try it; after all, practice may not make perfect PC boards, but after a while of trial and error you'll be making usable PC boards (see Fig. 3).

Fig. 4 illustrates a compact modular power supply layout that can be easily fabricated. The "U" shaped piece of aluminum should be made from 1/32 or 1/16 inch aluminum (yep, go get an aluminum door kick panel or heavy gauge cookie sheet). Placement of parts is not critical and there is enough room for whatever needs to be in the power supply. There is even enough room for a barrier terminal strip. Barrier terminal strip? Yep, if you use barrier terminal strips on power supplies, in your computer, and for connecting the real world to your computer, then it is very easy to disconnect a wire(s) from the barrier terminal strip with a screwdriver.

It is hoped that this short burst of ideas will motivate others into trying their hand at building some of the physical hardware of their system (terminal or computer). The ideas presented here can be realized with a little cash, some work, and perseverance. So if you are building your computer or terminal from scratch, partially from scratch, or from a kit, here are some ideas which should be of interest to you.

[1] *GC Electronics Printed Circuit Handbook* and several articles which have appeared in recent issues of *Popular Electronics* and *Radio-Electronics*.

Fig. 3. Typical 22-pin printed circuit board stock (minus printed wiring). Note: Gold plating on contacts is highly recommended; avoid surplus backplane sockets with any tarnish to interface with contact.

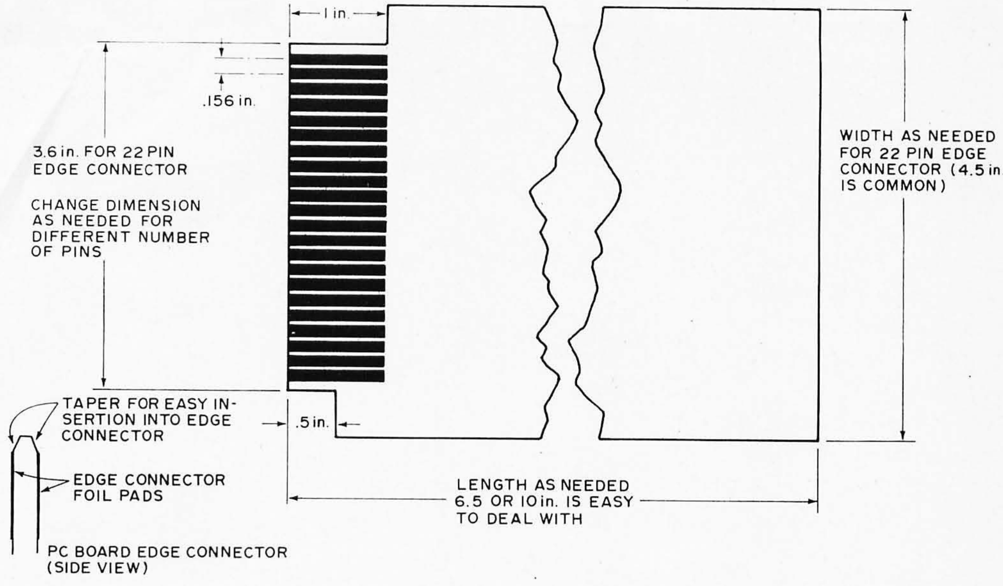
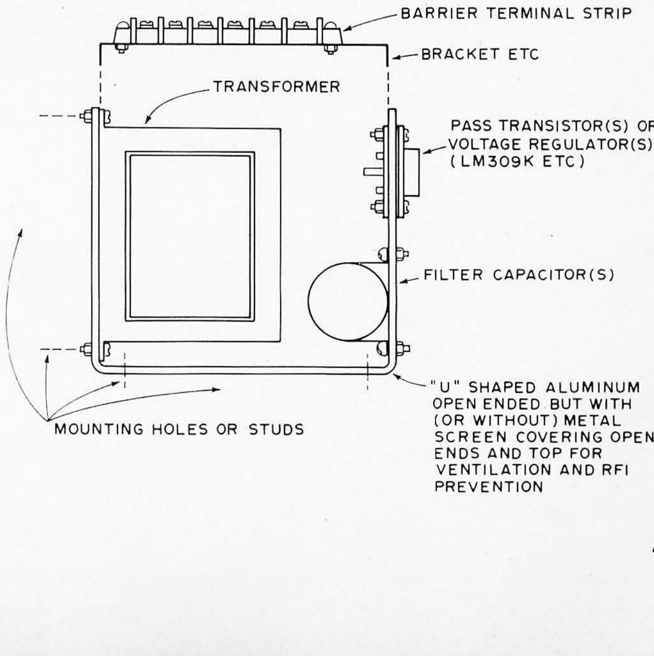
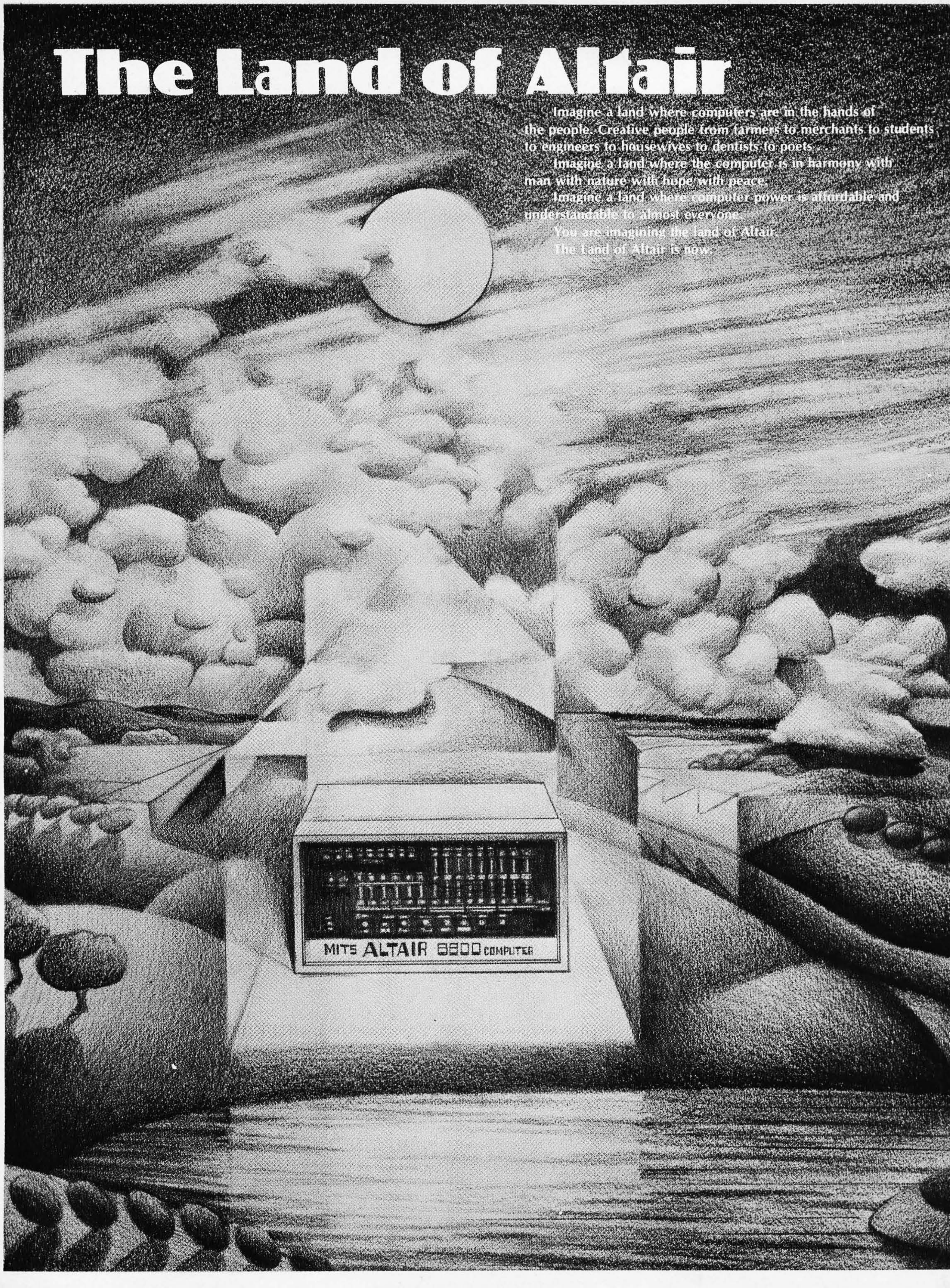
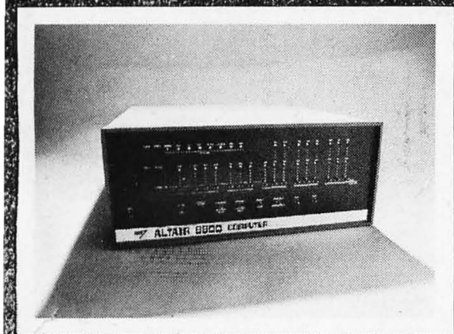


Fig. 4. Building a power supply as an assembly.




The Land of Altair
Imagine a land where computers are in the hands of the people. Creative people from farmers to merchants to students to engineers to housewives to dentists to poets . . .
Imagine a land where the computer is in harmony with man with nature with hope with peace.
Imagine a land where computer power is affordable and understandable to almost everyone.
You are imagining the land of Altair.
The Land of Altair is now.
MITS ALTAIR 8800 COMPUTER

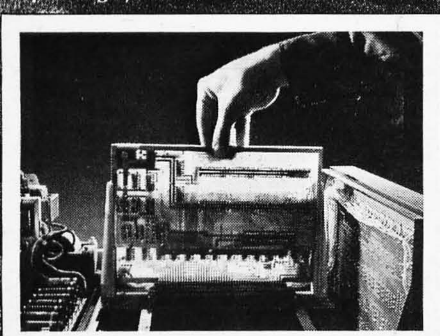


The Altair 8800 is a powerful, general purpose computer that sells for an amazingly low price of $439 in kit form and $621 assembled.

The Altair 8800 is a superbly engineered, variable word length computer. Its byte orientation structure was designed to give the Altair the most efficient utilization possible — an efficiency only found in the most advanced computers.

The Altair 8800 has bench marks comparable to those of much more expensive mini-computers. It has a cycle speed of 2 microseconds; it can directly address 65000 words of memory and 256 input/output devices; and it has over 200 basic machine instructions.

The Altair 8800 is backed by an extensive software development program. Altair BASIC and EXTENDED BASIC language are designed for most computer needs from business to home to scientific applications. Other software includes an Assembler, Text Editor, System Monitor, Debug, and Disk Operating System.



The Altair 8800 has been designed with buss orientation to be easily expanded and easily adapted to thousands of applications. Any card can be plugged into any slot and the correct address, etc., for that card will be picked up off the buss system.

The Altair can be custom assembled to fit almost anywhere. Besides general purpose computing (business, scientific and home uses), the Altair is ideal for process control and industrial uses.

The basic Altair includes Central Processing Board, front panel, power supply (enough to power any additional boards), and expander board (with room for 3 extra boards) all enclosed in a handsome, aluminum case. Up to 16 cards can be added inside the main case (memory and I/O boards).

Memory board options include 1024 word, 2048 word and 4096 word boards. Each memory board has memory protect features.

Interface board options include a Parallel Interface Board and 3 Serial Interface boards. These boards allow you to connect the Altair to our growing list of input/output devices.



For cost sensitive applications and for hobbyists, the Altair 8800 and most Altair options come in kit form. Already, thousands of Altair kits have been assembled and are in full operating order.

Altair kit builders include individuals, schools, companies, small laboratories, and industrial users.



The Altair Floppy Disk can store over 300,000 bytes of information on a flexible disk. With a data transfer rate of 250,000 words/second and a track to track access time of 10 msec., it has the capability for advanced data processing procedures.

Other Altair options include the Comter II Computer terminal with built-in audio-cassette interface for inexpensive mass storage, ASR-33 Teletypewriters, and the Altair 110 Line Printer which produces 80 columns of 5x7 dot matrix characters at 110 characters per second or 65 lines per minute.

PRICES:

Altair Computer kit with complete
assembly instructions $439
Assembled and tested Altair $621
1,024 word memory
card . . . .$97 kit and $139 assembled
2,048 word memory
card . . . .$145 kit and $195 assembled
4,096 word memory
card . . . .$264 kit and $338 assembled
Full Parallel Interface
card . . . .$92 kit and $114 assembled
Serial Interface card
(RS232) . . . . .$119 kit and $138 assembled
Serial Interface card (TTL or
teletype) . . . .$124 kit and $146 assembled
Expander Card (adds 4 slots
to 8800) . . . . . .$16 kit and $31 assembled

SOFTWARE PRICES:

Altair 4K BASIC $350
Purchasers of Altair 8800, 4K of Altair memory, and Altair I/O card .. ONLY $60
Altair 8K BASIC $500
Purchasers of Altair 8800, 8K of Altair memory, and Altair I/O card .. ONLY $75

Contact Factory for complete Altair Price List.

Altair Documentation Special . . . . . includes Assembly manual, Operators manual, Theory of Operation manual, BASIC language manual, catalog and sample Altair Users Group newspaper $15
(offer expires November 1, 1975)

Ordering Instructions

You can order the Altair Computer by simply filling out the coupon in this ad or by calling us at 505/265-7553 or 262-1951. Or you can ask for free technical consultation or for one of our free Altair System Catalogs.

Prices, specifications and delivery subject to change. Warranty: 90 days on parts for kits and 90 days on parts and labor for assembled units.

MITS
Creative Electronics

MITS/6328 Linn NE/Albuquerque, New Mexico 87108, 505-265-7553

ALTAIR COUPON

☐ Enclosed is check for $ ________
☐ BankAmericard # ________
☐ or Master Charge # ________
☐ Credit Card Expiration Date ________

☐ Altair 8800 ☐ Kit ☐ Assembled
☐ Options (list on separate sheet)
include $8 for postage and handling
☐ Altair Documentation Special
☐ Please send free Altair catalog

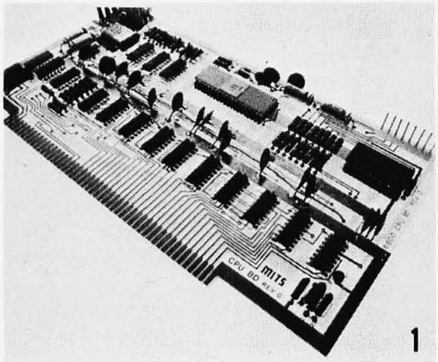
Name ________________________________

Address ________________________________

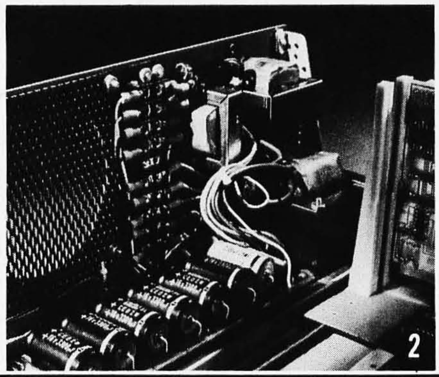
City ____________________ State & Zip ____________

MITS/6328 Linn NE, Albuquerque, NM 87108, 505/265-7553 or 262-1952

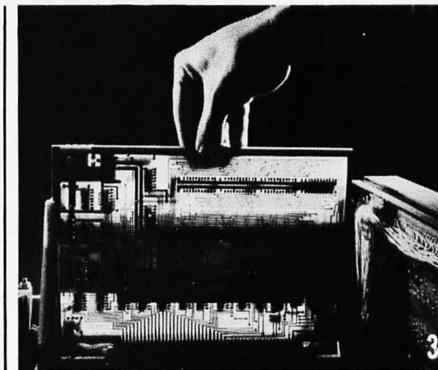
INSIDE the Altair® Computer



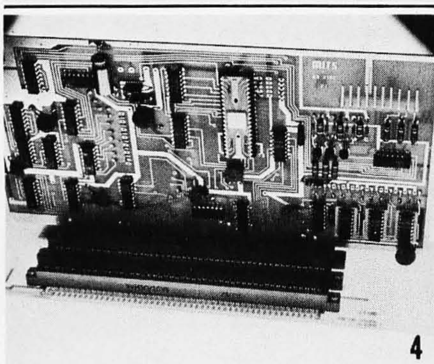
1



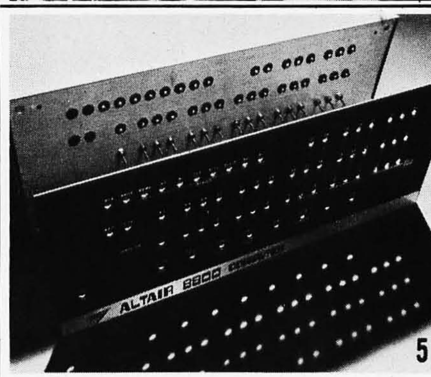
2



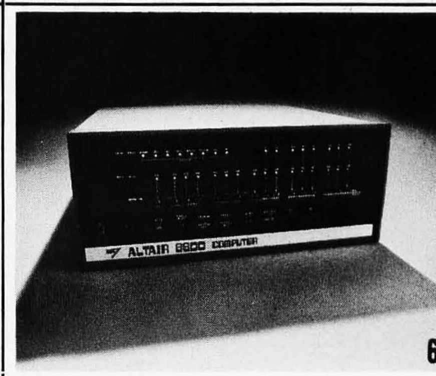
3



4



5



6

1. Central Processing Unit (CPU) Board. This double-sided board is the heart of the Altair 8800. It was designed around a powerful, byte oriented, variable word length processor—a complete central processing unit on a single LSI chip using n-channel, silicon gate MOS technology. The CPU board also contains the Altair System Clock—a standard TTL oscillator with a 2.000 MHz crystal as the feedback element.

2. Power Supply. The Altair Power Supply provides two +8, a +16 and a −16 volts. These voltages are unregulated until they reach the individual boards (CPU, Front Panel, Memory, I/O, etc.). Each board has all the necessary regulation for its own operation.

The Altair Power Supply allows you to expand your computer by adding up to 16 boards inside the main case. Provisions for the addition of a cooling fan are part of the Altair design.

3. Expandability and custom designing. The Altair has been designed to be easily expanded and easily adapted to thousands of applications. The basic Altair comes with one expander board capable of holding four vertical boards. Three additional expander boards can be added inside the main case.

4. Altair Options. Memory boards now available include a 1024 word memory board, a 2048 word memory board, and a 4096 word memory board. Interface boards include a parallel board and 3 serial boards (RS232, TTL and Teletype). Note: Interface boards allow you to connect the Altair Computer to computer terminals, teletypewriters, line printers, plotters and other devices.

Other Altair Options include additional expander boards, computer terminals, audio-cassette interface board, line printers, ASCII keyboards, floppy disc system, alpha-numeric display and more.

5. All aluminum case and dress panel. The Altair Computer has been designed both for the hobbyist and for industrial use. It comes in an all aluminum case complete with sub-panel and dress panel.

6. It all adds up to one fantastic computer. The Altair is comparable to mini-computers costing 10-20 thousand dollars. It can be connected to 256 input/output devices and can directly address up to 65,000 words of memory. It has over 200 machine instructions and a cycle time of 2 microseconds.

You can order the Altair Computer by simply filling out the coupon in this ad or by calling us at 505/265-7553. Or you can ask for free technical consultation or for one of our free Altair System Catalogues.

PRICES:

Altair Computer kit with complete assembly instructions $439.00
Assembled and tested Altair Computer $621.00
1,024 word memory board .... $97.00 kit and $139.00 assembled
2,048 word memory board .... $145.00 kit and $195.00 assembled
4,096 word memory board .... $264.00 kit and $338.00 assembled.
Full Parallel Interface board .... $92.00 kit and $114.00 assembled.
Serial Interface board (RS232) .... $119.00 kit and $138.00 assembled.
Serial Interface board (TTL or teletype) .... $124.00 kit and $146.00 assembled
Audio Cassette Record Interface ... $128.00 kit and $174.00 assembled
Expander Board (adds 4 slots to 8800) ... $16.00 kit and $31.00 assembled

NOTE: Altair Computers come with complete documentation and operating instructions. Altair customers receive software and general computer information through free membership to the Altair User's Club. Software now available includes a resident assembler, system monitor, text editor and BASIC language.

MITS/6328 Linn NE, Albuquerque, NM, 87108 505/265-7553

MITS

"Creative Electronics"

Prices and specifications subject to change without notice. Warranty: 90 days on parts for kits and 90 days on parts and labor for assembled units.

MAIL THIS COUPON TODAY!

☐ Enclosed is check for $________
☐ BankAmericard #________
☐ or Master Charge #________
☐ Credit Card Expiration Date ________
☐ Altair Computer ☐ Kit ☐ Assembled
☐ Options (list on separate sheet)
Include $8.00 for postage and handling.
☐ **PLEASE SEND FREE ALTAIR SYSTEM CATALOGUE**

NAME________
ADDRESS________
CITY________ STATE & ZIP________

MITS/6328 Linn NE, Albuquerque, NM, 87108
505/265-7553

Permission Granted to Xerox

(or 3M, IBM, Apeco or otherwise facsimile, copy or duplicate)

BYTE SUBSCRIBERS WILL LEARN . . .

About all of the new small systems coming on the market . . . which are real and which are still in development stages . . . there will be a simple series of articles on computer languages . . . reviews of kits . . . info on surplus buys and used equipment . . . circuits for building your own equipment . . . discussions of standards . . . news of user clubs and programs available . . . info on specialized newsletters . . . games and game groups . . . game programs . . . computer art . . . computer music . . . and if that doesn't cover it, all you have to do is write in and either say what else you want or else start writing the articles yourself.

BYTERS SAVE . . .

With single copies going for $1.50 (insane price), the $12 subscription price begins to look attractive. You won't want to miss a copy (presumably), so fill in the coupon here and send it in quickly . . . no cash needed, just send it in, we'll put you into the subscription computer (with fingers crossed) and send you the invoice. It's probably a business expense, right? If you dislike ripping your soon to be valuable issue of BYTE to shreds by removing a coupon, you are hereby given permission in perpetuity (until subscription rates go up) to violate the copyright and make a copy of any one of the four coupons . . . neatness counts. Your friends may do likewise, unless you are a lot less persuasive than we think you are.

The Management

Cut

Name . . . .

Address . . . .

City . . . . State . . . . Zip . . . .

☐ BILL ME ☐ Check for $12 enclosed
☐ Bill BankAmericard or MasterCharge # . . . .

Tear

Name . . . .

Address . . . .

City . . . . State . . . . Zip . . . .

☐ BILL ME ☐ Check for $12 enclosed
☐ Bill BankAmericard or MasterCharge # . . . .

Shred

Name . . . .

Address . . . .

City . . . . State . . . . Zip . . . .

☐ BILL ME ☐ Check for $12 enclosed
☐ Bill BankAmericard or MasterCharge # . . . .

Rip

Name . . . .

Address . . . .

City . . . . State . . . . Zip . . . .

☐ BILL ME ☐ Check for $12 enclosed
☐ Bill BankAmericard or MasterCharge # . . . .

Add a Stack to Your 8008

by
Hal Chamberlin
The Computer Hobbyist
PO Box 295
Cary NC 27511

Besides higher speed, the most significant improvement offered by the 8080 is the addition of a general purpose stack capability. Using the stack, the programmer may save registers used in subroutines and interrupt service routines and then later restore them. Arguments to subroutines may also be pushed onto the stack. In the 8080 the stack is kept in main memory and addressed by means of a stack pointer register. One inconvenience of the 8080 stack is that data may be pushed and popped only in byte pairs creating wasted space if only a single register needs to be saved. Also, the stack pointer MUST be set up at the beginning of a program before any subroutines are called and kept valid at all times or very strange things may happen.

Much of the programming convenience of the 8080 stack may be had on an 8008 system with the addition of about six ICs and the use of one input and one output address. The basic stack is 16 elements deep which is generally adequate for register saving applications. Addition of more chips and substitution of 256 X 4 RAMs for the 16 X 4 RAMs gives a 256 element capacity, ample for almost any use. In either case the added hardware provides both a stack pointer and a dedicated memory. The stack pointer does not need to be initialized and thus the stack is always ready for use, or it can be completely ignored by programs that don't need it without problems.

Programming the data stack is quite simple. The output address associated with the stack is given the symbolic name STPSH for STack PuSH and the input address is given the name STPOP for STack POP. When an OUT STPSH is executed by the program, all of the existing data (or garbage) in the stack is conceptually pushed down one location and the byte in register A is written into the top location which was vacated. When an INP STPOP is executed, the contents of the top location are read into register A and then all of the lower data in the stack moves up one location and the top location is lost.

An obvious application of the stack is in writing subroutines that do their job without destroying any registers. A simple example is the exchange HL and DE subroutine in Fig. 1. First register A is pushed onto the stack. Then registers H and D and L and E are exchanged using A. Finally the original state of register A is restored by popping it off the stack and the subroutine returns. Because of the push-down nature of the stack, one subroutine that uses the stack may call another subroutine

Fig. 1. A subroutine to exchange DE and HL register pairs using one stack location and no additional registers.

```
XCDEHL OUT  STPSH     SAVE A ON THE STACK
       LAH            EXCHANGE H AND D
       LHD            USING A
       LDA
       LAL            EXCHANGE L AND E
       LLE
       LEA
       INP  STPOP     RESTORE A FROM STACK
       RET            AND RETURN
```

Reprinted by permission from The Computer Hobbyist, *May 1975.*

Hal Chamberlin and his associates at *The Computer Hobbyist* put out excellent small systems technology . . . designs include a high reliability audio cassette recording method, an inexpensive high resolution graphics display – and this article's stack design among others. Several of their more general purpose designs (e.g., tape interface, CRT display) are soon to be available in kit or assembled versions. This article describes a custom modification of an 8008 based system which you can add to an input/output port to achieve a stack mechanism. The method is that old standby of minicomputer instruction set escape mechanisms – use I/0 commands to implement "new instructions." With a stack of sufficient size and suitable save/restore subroutines accessed by RST instructions of the 8008, you can eliminate conflicts in register usage between multiple levels of subroutines. The overhead penalty is a single RST or CAL instruction in the linkage code, the register save and restore routines, and the time required to execute the save/restore subroutines.

. . . CARL

This circuit brings the 8008 one step closer to the goal of a "real" computer.

that uses it and so on without loss of data as long as the stack's capacity is not exceeded. The only requirement is that all of the data saved on the stack by a subroutine be popped back off before it returns.

Fig. 2 shows a completely general interrupt service routine that uses the data stack to save all registers and the state of the conditions (C, Z, S, P). When entered, register A is first pushed onto the stack. Then the remaining six registers are saved one at a time by first loading them into A and then pushing A onto the stack. None of the instructions necessary to do this affect the condition flags. Finally a chain of conditional jumps is executed to create a "magic number" in A that reflects the state of the conditions. After A is pushed onto the stack, the interrupt may be serviced without restrictions on register usage.

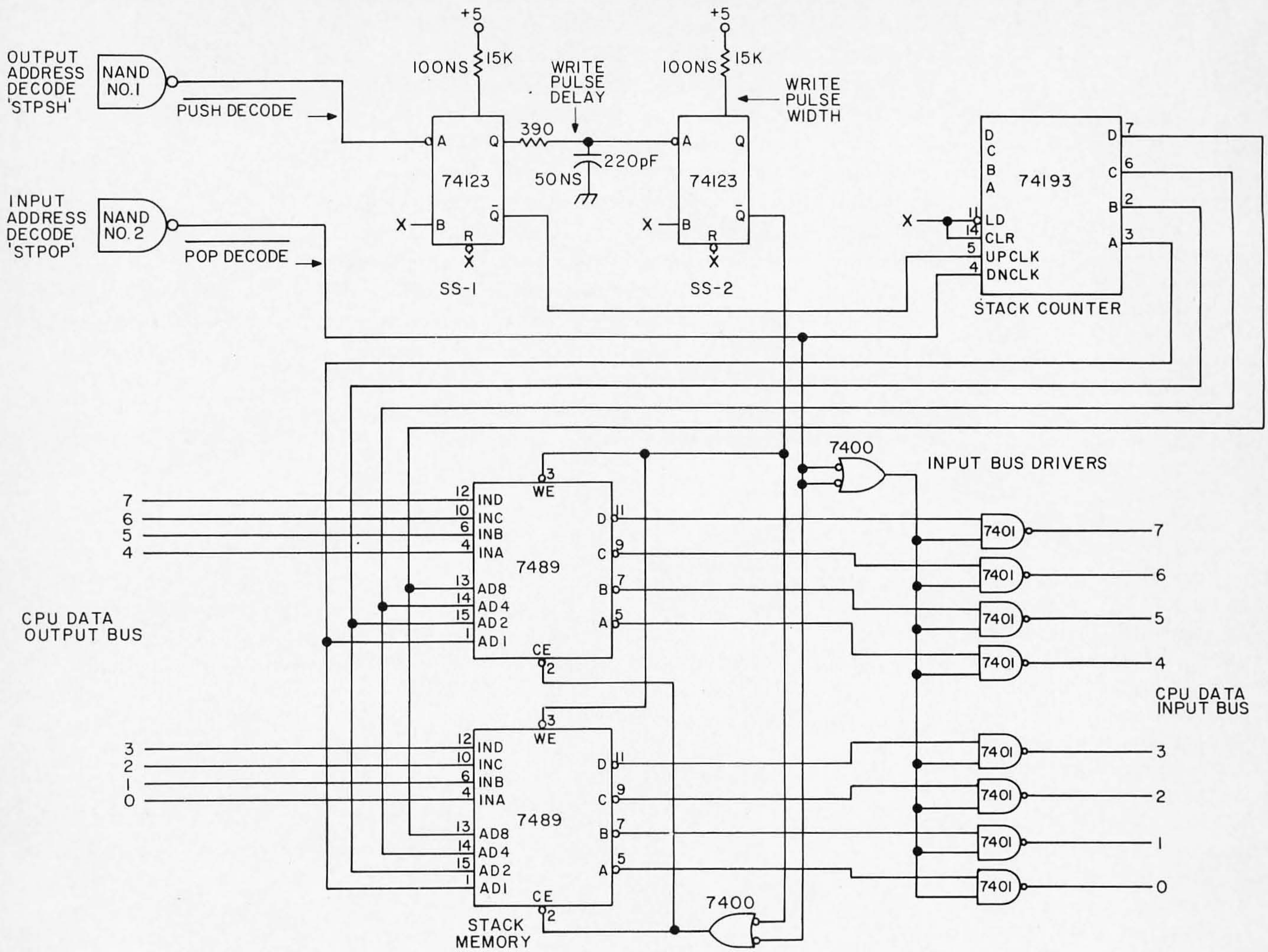
In order to return to the interrupted program, first the magic number is popped off and added to itself with an ADA instruction. The number is such that the proper conditions are set when it is added to itself. Next the six index registers are popped off and restored in reverse order from which they were saved. Finally A is restored and a RET instruction is executed. This method of complete status saving may be modified for use by a debug program. Debug can be entered by a console interrupt and the user may examine things. Then program execution may be resumed with no loss of data. These two programming examples should serve to illustrate the use of the stack.

The stack is implemented with an up-down counter and

Fig. 2. A general purpose register and condition code save routine.

```
GSAVE  OUT  STPSH    SAVE A ON THE STACK
       LAB           SAVE B
       OUT  STPSH
       LAC           SAVE C
       OUT  STPSH
       LAD           SAVE D
       OUT  STPSH
       LAE           SAVE E
       OUT  STPSH
       LAH           SAVE H
       OUT  STPSH
       LAL           SAVE L
       OUT  STPSH
       LAI  0        CLEAR A
       RAR           PUT CARRY IN HIGH ORDER
       JTZ  GSAV3    JUMP IF ZERO FLAG IS ON
       LBI  170B     PUT INTO B THE BIT MASK TO
       JTS  GSAV1       TURN OFF THE ZERO FLAG AND
       LBI  030B        RESTORE THE SIGN FLAG
GSAV1  JTP  GSAV2    OR IN A 004B IF PARITY
       ORI  004B        INDICATOR IS OFF
GSAV2  ORB           COMBINE B AND A
GSAV3  OUT  STPSH    SAVE MAGIC NUMBER ON STACK
  *         REGISTER AND CONDITION RESTORE ROUTINE
GRSTR  INP  STPOP    RESTORE MAGIC NUMBER FROM STACK
       ADA           ADD IT TO ITSELF TO RESTORE CONDITIONS
       INP  STPOP    RESTORE L
       LLA
       INP  STPOP    RESTORE H
       LHA
       INP  STPOP    RESTORE E
       LEA
       INP  STPOP    RESTORE D
       LDA
       INP  STPOP    RESTORE C
       LCA
       INP  STPOP    RESTORE B
       LBA
       INP  STPOP    RESTORE A
       RET           RETURN WITH STATUS RESTORED
```

Fig. 3. Logic Diagram of a 16 element data stack. NOTE: "X" refers to a source of logical one, usually a 1k resistor to +5.



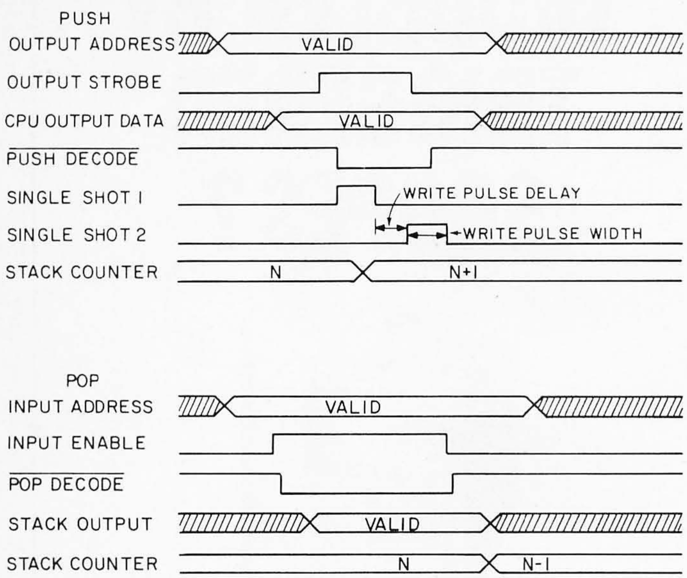
An obvious application of the stack is in writing subroutines that do their job without destroying any registers.

a random access read-write memory. Rather than the data moving when pushes and pops are executed, the up-down counter acts as a pointer to the top element on the stack and the pointer moves. The logic is set up so that when an OUT STPSH is decoded, the counter first counts up one notch and after sufficient time for the address to settle in the RAM, a write pulse is generated to write the data from A into the RAM. The write pulse delay can be fairly short (50 NS or so) in the 16 element stack but must be at least 200 NS for the slower MOS RAM used in the 256 element version. It is possible that a timing problem may arise in a system using the 8008-1 if the output data is not valid for the sum of write pulse delay and write pulse width (950 NS) required by the MOS RAM. (Timing given is for the 2101 RAM. Matters are improved if 2101-1, 2101-2, or 9101 RAM is used.) There should be no problems with the bipolar RAM in the 16 element version.

When an INP STPOP is recognized, the contents of the currently addressed location are simply gated onto the input bus. The counter counts down one notch at the end of the INP instruction thereby addressing the next lower element on the stack.

Figs. 3 and 4 show the logic diagram and timing chart respectively for a 16 element data stack. A bus type of I/O system (as opposed to a "port" type) is assumed. As shown, any system with either separate data input and output busses or a bidirectional bus may be used. Some systems may use an output bus with TRUE data and an input bus requiring FALSE data. In this case, the 7401s may be omitted and the TTL RAM outputs tied directly to the input bus. The two single-shots, SS-1 and SS-2, are used to time the sequence

Fig. 4. Stack Timing Diagram.



of events for a stack push. First, NAND gate number 1 recognizes the coincidence of the STPSH device code on the address bus and an output strobe pulse or its equivalent. The gate output triggers SS-1 which increments the stack pointer counter when its cycle is finished. An RC network between the two single-shots delays firing of SS-2 until the counter has settled down and the RAMs recognize the new address. The write enable is connected to SS-2 which allows data on the CPU output bus to be written into the newly addressed RAM location.

The occurrence of an INP STPOP is detected by NAND gate number 2. As long as the gate is satisfied, data from the RAM is placed on the CPU input bus. At the end of the INP instruction when the NAND gate output goes back to a ONE, the counter decrements to address the next lower element on the stack. A 7400 connected as an OR-NOT enables the memory when either a push or a pop is being executed and disables it otherwise.

The logic necessary for a 256 element stack is essentially the same as for the 16 element version. The major differences are that a separate single-shot should be used to time the write delay and that a buffer is absolutely necessary to drive the CPU input bus. If the polarity of the input bus is the same as that of the output bus or it is the same bus, 8093 or 74125 noninverting and quad tri-state buffers are convenient to use. Open-collector 7401 gates may be used instead if the input bus is inverted. The chip enables on the MOS RAMS should be grounded so that the chip is always enabled. The connection to the bus drivers is left as it was for the 16 element version however. The timings for the write delay and write pulse width single-shots can be set to the minimum values allowable for the standard 2101 RAM. If a 9101 is used, the timing may be speeded up considerably. An 8101 may require somewhat slower timing. In any case be sure to check the data sheet for the RAM being used.

After writing a few programs using the stack you will wonder how you got along without it. The size and speed of some routines may be improved by a factor of two if use of the stack alleviates the need to constantly reference memory. An overall improvement of 10 to 20% can be expected on large programs such as assemblers. The biggest improvement however will be in coding time since register usage will not have to be carefully planned in advance.

Use the stack to pass parameters to subroutines when you don't have enough registers.

THE CURVE TRACER THAT WON'T COLLECT DUST.



The Hickok Model 440 semiconductor curve tracer is all purpose and convenient to use. It's the ideal instrument for testing, evaluating, classifying and matching all types of transistors, FET's and diodes. You'll get stable, full range dynamic displays that you can accurately scale right from the screen.

- Pull-out card for easy, fast set-up and operation.
- Set-up marks for rapid set-up of 80% of tests.
- Unique INSTA-BETA display takes the guesswork out of transistor and FET parameter measurement.
- In-or-out of circuit testing.
- A full range professional tracer at a price you can afford.

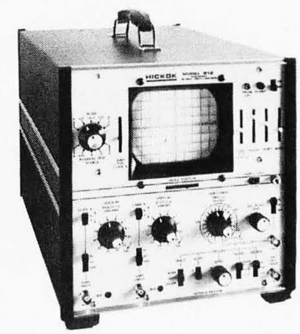
AT YOUR DISTRIBUTOR **$165.00**

HICKOK

the value innovator

INSTRUMENTATION & CONTROLS DIVISION
THE HICKOK ELECTRICAL INSTRUMENT CO.
10514 Dupont Avenue • Cleveland, Ohio 44108
(216) 541-8060 • TWX: 810-421-8286

TROUBLED BY TRIGGERED SCOPES?



The Hickok Model 512 Dual Trace Oscilloscope eliminates the set-up and precision problems you've had to accept using other triggered scopes.

It's easy to set up

- Simplified color-coded front panel controls.
- Beam finder quickly locates off-scale traces.
- Foolproof triggering to 15 MHz.

It gives you superior performance

- 10 MHz response flat within 3dB. Excellent pulse response.
- 3% accuracy on vertical and horizontal ranger.

Hickok industrial lab quality and construction

Glass epoxy PC boards used throughout. Regulated power supply.

AT YOUR DISTRIBUTOR **$675.00**

complete with probes and accessories

HICKOK

the value innovator

INSTRUMENTATION & CONTROLS DIVISION
THE HICKOK ELECTRICAL INSTRUMENT CO.
10514 Dupont Avenue • Cleveland, Ohio 44108
(216) 541-8060 • TWX: 810-421-8286

The Shadow, Buck Rogers, and the Home Computer

by
Richard Gardner
Box 134
Harvard Square
Cambridge MA 02139

A computer at home? Ask many present day computer systems people what they'd do with a home computer and you'll get the old silent treatment in return. But all that indicates is a lack of imagination. A large part of the BYTE philosophy is the discovery of applications areas through the imaginations and practical results of readers. Richard Gardner supplies us with a "Gee Whiz" article on potential applications areas to get things in motion a bit. Richard has extensive computer applications experience including one stint working for the Children's Museum in Boston, creating interactive computer oriented exhibits. Eventually, many of the systems ideas Richard mentions in his article will appear as practical plans and programs in the pages of BYTE – as developed and described by our readers. If you'd like an interactive meeting of the minds on possible uses and ideas, Richard invites correspondence from readers. ... CARL

Ah yes! It conjures up visions of an earlier day, many years ago, when Mom, Pop and the kids sat around that newfangled gadget, the radio, and listened to "The Shadow" and "Buck Rogers."

Flash forward to the future, right now! Again, we see Mom, Pop and the kids sitting around that newfangled gadget, the computer, balancing a checkbook, converting a four servings recipe to seven, and playing tic-tac-toe. Not very exciting things to do with a computer, you say? Well, you're right. But let's see if we can do something to make it at least as exciting as old-time radio.

We mentioned three applications for a home computer:

1) checkbook balancing
2) recipe converting
3) game playing

For starters Mom and Pop should have a program for collecting and summarizing all their financial data, on a daily, monthly and yearly (for your "friend" and mine, the IRS) basis. A family will be more secure by knowing the state of its financial affairs. You will want to



compute interest for different purchase plans, and balance the checkbook.

Moving on to a subject close to my heart (just below, and a little to the right) – food. Almost anyone can convert 4 to 7 servings – just double it and feed the leftovers to the dog, or give it to a charitable organization (tax deductible, of course). What you really want to know is whether everyone got enough nutrients (vitamins, minerals, protein, etc.) from what they ate today. Hint: it can be done. I know of two people who started a small company to do it.

On to fun and games – hundreds of game playing programs have been written (I invented one called YOUGUESS) for all sorts of computers and languages. You should have them all. It will win friends and influence neighbors, if you'll pardon the pun.

I'd say that's at least as exciting as old-time radio. Good, but we can do much better. Let's consider three things:

1) Today's computers are very fast. The applications we've mentioned *might* take one hour of CPU time per day, at the very most. So what do we do with the other 23+ hours?

2) There are lots of computers in the world, and they can talk to each other.

3) Computers can hear, see, feel, smell and touch.

Keep these things in mind as we consider what might be called economic, personal and educational applications for the home computer.

Computerizing the Home

Since your computer won't be doing anything most of the day why not put it to work:

1) *Heating and air conditioning control.* Optimize increases and decreases in the inside temperature to minimize energy use. Open and close curtains on windows to use the sun's energy or keep it out.

2) *Security.* While you're at home or away, monitor the opening and closing of windows and doors. Automatically telephone the police with a recorded message when you're gone or at home. Monitor the use of your swimming pool – sound an alarm when the pool is in use and nobody's in the lifeguard seat. Fire monitoring equipment can be located in many places and sound an alarm long before you might smell or see smoke. The fire department can be called automatically with another recorded message.

These applications will make use of photocells, theramins (motion sensing devices), heat sensors, contact switches, smelling devices (like those used by the Defense Department in Vietnam to smell passing elephants and tigers). Eight bits might be used to represent a temperature range of 256 degrees. 100 degrees would be adequate for most locations. One analog to digital converter could be used for other analog inputs, such as from a photocell. A digital to analog converter would generate voltages to be used by motors and other mechanisms.

Using a Symbol Table to Improve the Food Table

Most people in America have a poor diet in spite of the fact that we have more food of a better quality and variety than any other country. So I consider the following to be important uses for a home computer:

1) Selection of foods on a seasonal basis to reduce cost and improve quality. A program for doing this would run for a year and use a data base for your area (to take advantage of local produce). A second data base would be programmed for widely available foods and when they are best and cheapest.

2) A menu building program to take advantage of the above system but with the intention of increasing variety and maximizing nutrition.

3) A shopping guide to take advantage of local food supplies by indicating the best one or two markets from which to purchase your food. This data base might be maintained by some person in your locality – and then rented on a per use basis. No sense in everyone typing in today's price for cumquats. Perhaps the *New York Times* will eventually computerize its cookbook, plus thousands of other recipes, and allow the public to access this data base via a personal computer.



Since your computer won't be doing anything most of the day . . .

This application, like others mentioned, would use the telephone system – the world's largest computer. I can see it now. The kids get home from school and ask, "What's for dinner, Ma Bell?"

The Bottom Line Isn't Always an End Statement

Or, how to profit from your home computer:

1) Income management, as previously mentioned, but with the help of another computer. Several computer companies that do nothing but figure taxes (for you know who!) already exist. Eventually they will allow your computer to call their computer. Your computer shovels in a year's data and out pops a tax form with all the right numbers. You might think it easier to do your own programming, but remember that you can't write every program you will want to use. In addition, these companies have staffs that do nothing but make program improvements and changes required by the IRS. What person in his or her right mind could possibly keep track of a myriad of new rules from the IRS?

2) Play the ponies or the puppies? An obvious use for your computer. Again, use a data base compiled by some local eager beaver. Perhaps you'd be charged a small fee for accessing the day's statistics. Perhaps you have a data base or program to trade.

3) Then there's always the world's biggest daily crap



game – the stock market. A company in Philadelphia will charge you $300 a year for a small numeric terminal and 24 hour a day access to their stock data base. You key in the number of a stock and out pops the high, low, average, etc. Your computer could make one call after each trading day, collect the stock data you're interested in, hang up, and then determine if you should buy, sell or hold. The decision making could be done by your program or one being rented from a stock market wizard you know.

4) I mentioned how a computer could be used to optimize the purchase of food. This principle applies to any commodity whose price and quality changes during the year: clothes, home furnishings, gifts, transportation, even housing. Some local person, or you, could create the necessary commodity and price data bases, then use or rent them.

Remember! There is a host of areas for small business activities using your home computer as a tool of the trade. All it takes is imagination, a bit of digging into the wants and concerns of your neightbors, and the programming of your computer.

Six Munce Ago I Couldn't Even Spell Computer Programmmer . . .

Computers are good for keeping you in touch with the world. For example:

1) The *New York Times* has a computerized data base of all its back issues – currently accessible to the general public, for a fee. The cost will probably go down to the point where you might program your computer to query the *Times* data base and retrieve front page stories, financial page stories, or any story that contains a keyword or some combination of keywords. This would be done early in the morning and read by you at breakfast time.

2) Your local university or high school might have a computer with courses that can be taken via a remote terminal. Many universities already give some courses using only this method.

3) The Children's Museum in Boston will eventually allow you to call their computer, via a terminal or computer, and access a data base of cultural, educational, and social events in the Boston area. Your computer might call theirs once a day to learn what's new or learn about a particular type of event.

Computers As Toys

Computers are probably the greatest toy ever invented. Here are some examples of how you can play around with yours:

1) It has been rumored that 50%, or more, of the computer time used at MIT is used to play Space War – the Grandpa of computer games! Your computer, a TV set, a few buttons and switches and, presto – Space War! Or ping-pong, or driving down a road, flying and landing an airplane, landing on the moon, chess, checkers (you can play these games in Boston with the Children's Museum computer).

2) Toys that play with you – like robots. The Boston Children's Museum has a robot that was built for about $200. Mass production of a special chip and board will bring that cost down. Then the biggest cost will be the Meccano Set (like an erector set, only better), which can be used to build almost any sort of mechanical device. How about a robot to do housework?

3) The ultimate fun, though, is to write your own programs to do all these things! Kids, and adults, will play only so many games of tic-tac-toe – then they want to know how it works. Help them write their first BASIC program . . . and they're likely to be hooked for life! Eventually programming will include a broader range of input/out devices such as the previously mentioned buttons and switches, photocells, microphones, etc. This will lead to the applications just discussed, and who knows what?

These are just some of the possible applications for a home computer. All of them might not be reasonable or practical things to do but they should set you to thinking.

As future issues of BYTE unfold, the Gee Whizzers applications will lead to practical articles on the software and specialized peripherals needed to implement some of these ideas.

from the Publisher . . .

are they real?

We have a lot of buzz words these days, and one phrase we hear a lot is about companies being "real". Being the publisher of BYTE, it seemed to me that it would be worthwhile if I were to make a trip to visit the major microcomputer systems companies and talk with them ... possibly making them more real to our readers.

My first stop was in Denver – a short layover between planes on my way to Salt Lake. I tried to locate the Digital Group there, but had no luck. Sorry about that. I know they are real because I recently sent them a check for their video display generator and received one a few days later in the mail. I was disappointed not to have a chance to talk with them at greater length.

In Salt Lake I was met by Doug Hancey of Sphere and driven out to their new plant in Bountiful, a suburb. It's a small building and I have a feeling that they will quickly be outgrowing the facilities ... I don't think they realize what a demand there is going to be for their system.

They had a prototype up and running and it looked good. There were a couple of glitches, of course, but they seemed to have these well in hand and were expecting to be able to set up a production department very soon. The system is based upon the Motorola M6800 chip and features a PROM loader. They plan to have another PROM with Basic in it, which would be quite a step ahead for it would free all of the RAM memory for use and would permit instant use of Basic without the usual loading process.

I gather that Motorola has been extremely helpful in supplying information and support for the effort. This may have a lot to do with

Continued on page 81

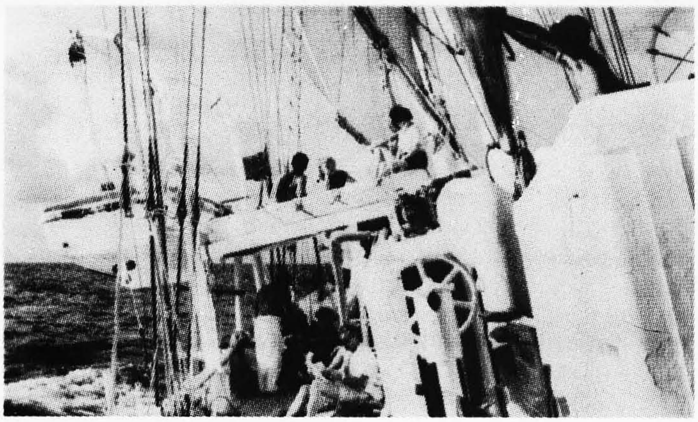


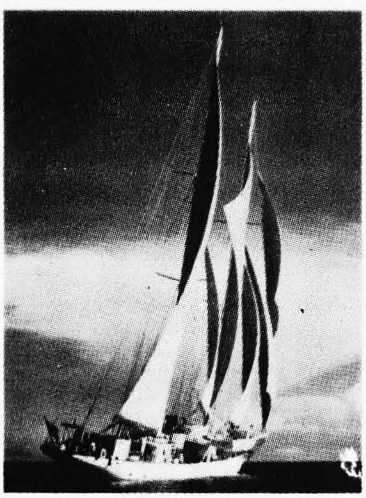
Sphere HQ in Bountiful, Utah, just outside of Salt Lake. That's Michael Wise, the president, in front. These are new offices for Sphere and the production will be set up in the back part of the building.

BYTE staff

EDITOR
Carl T. Helmers Jr.
PUBLISHER
Wayne Green
MANAGING EDITOR
Judith Havey
ASSOCIATE EDITORS
Dan Fylstra
Chris Ryland
CONTRIBUTING EDITORS
Hal Chamberlin
Don Lancaster
ASSISTANT EDITORS
John C. Burnett
Susan G. Philbrick
PRODUCTION MANAGER
Lynn Panciera-Fraser
ART DEPARTMENT
Nancy Estle
Neal Kandel
Peri Mahoney
Bob Sawyer
PRINTING
Biff Mahoney
PHOTOGRAPHY
Bill Heydolph
Ed Crabtree
TYPESETTING
Barbara J. Latti
Marge McCarthy
ADVERTISING
Bill Edwards
Nancy Cluff
CIRCULATION
Susan Chandler
Dorothy Gibson
Cheryl Hurd
Pearl Lahey
Lorraine Morton
Judy Waterman
INVENTORY CONTROL
Marshall Raymond
DRAFTING
Bill Morello
COMPTROLLER
Knud E. M. Keller

Take off your shoes.







Hit the deck in shorts and a tee shirt. Or your bikini if you want.

You're on a leisurely cruise to remote islands. With names like Martinique, Grenada, Guadeloupe. Those are the ones you've heard of.

A big, beautiful sailing vessel glides from one breathtaking Caribbean jewel to another. And you're aboard, having the time of your life with an intimate group of lively, fun-loving people. Singles and couples, too. There's good food, "grog," and a few pleasant comforts...but there's little resemblance to a stay at a fancy hotel, and you'll be happy about that.

Spend ten days exploring paradise and getting to know congenial people. There's no other vacation like it.

Your share from $245. A new cruise is forming now. Write Cap'n Mike for your free adventure booklet in full color.

Windjammer Cruises.
A WINDJAMMER INTERNATIONAL SUBSIDIARY • OTC

Name ______________________________

Address ______________________________

City __________ State __________ Zip __________

Phone ______________________________

P.O. Box 120, Dept. 121 Miami Beach, Florida 33139

A NOVAL Assembler for the 8008 Microprocessor

by
Peter H. Helmers
Box 6297
River Station
Rochester NY 14627

To the hardware specialist who has just spent months building his own personal computer (well, in the case of the 8008, his own personal processor), having to actually program the computer can be an entirely new and unique experience. In the author's case, writing an audio cassette storage based operating system for the 8008 was fun. At least it was fun the first time that the program was hard assembled and the first time that some unique, sophisticated and entirely ambiguous method of double indirect addressing had to be developed. However, when it came time to rewrite the code for the operating system, all of this had to be done again: Every byte of code had to be assigned to its core location; every label had to be defined; and every time double indirect addressing or the equivalent was required, the code had to be written out byte by byte. Well, there had to be a better way.

A "Noval" Concept

The solution adopted is not unique except in its implementation: Use a piece of software called a macro assembler. One way to accomplish this is to write a macro assembler which can run on a minimal 8008 system. However, despite doubts about whether this is even theoretically possible with the 8008, writing a macro assembler is like re-inventing the wheel (which in itself was a "revolutionary" idea . . .). Just about every large, medium and small scale computer has a macro assembler. Wouldn't it be nice to adapt one to the 8008? It turns out that the Data General Corp. Nova Minicomputer has a macro assembler which is very easy to adapt to the 8008.

What Is An Assembler?

Computers, being digital machines, operate on a series of numeric codes which instruct them to perform one of a fixed set of operations. People, on the other hand, find it easier to instruct the computer using symbols rather than numeric codes. For example with the 8008, HLT (e.g., stop!) is easier to conceptualize and understand than just: 377_8. The process of translating symbols into a numeric code is called an assembly. The software that does this is called an assembler.

In the simplest case, there is a one to one correspondence between the symbols used by the human programmer and the numeric codes used by the computer. Basically, in order to use the Nova macro assembler to assemble programs using 8008 mnemonics, all that has to be done is to redefine the assembler's symbol table so that the correct mapping between 8008 symbols and numeric codes is achieved.

The symbol table for the DGC macro assembler is generally contained in a file named MAC.PS. However, Data General has made it very easy to redefine any and/or all of the symbols in the MAC.PS file. This is done by using two special operands that the macro assembler recognizes: .XPNG and .DUSR. The .XPNG operand (meaning "expunge") deletes

I wonder how many readers have access to one or more minicomputer systems — or larger systems. It might be a question of how many readers have already been in the "pond" of using computers in one form or another versus how many are just in the process of "diving in" for the first time. For those readers who have access to a mini, the software development aspects of the home microcomputer can be simplified by use of "cross assembly" techniques. In this article Peter Helmers tells how he and Loren Woody implemented the NOVAL assembler while undergraduates at the University of Rochester. NOVAL is an 8008 cross-assembler running on a Data General NOVA, based upon extensions of the Data General Macro Assembler.

. . . CARL

all Nova symbols from the symbol table. The .DUSR operand (meaning "define user symbol") can then be used to equate most of the 8008 mnemonics to corresponding numeric codes. The use of these two operands can be seen in the listing of the INTEL.PS file shown in Fig. 1.

The .XPNG operand first deletes the Nova mnemonics from the MAC.PS symbol table. Then after telling the macro assembler that all numbers are interpreted as octal numbers (by means of the .RDX and .RDXO operands), the line:

.DUSR ACA=210

tells the assembler to insert an entry in its symbol table that will cause the mnemonic ACA (e.g., add register A to register A with carry) to be translated into the octal code 210 every time that the ACA operand is encountered in the 8008 source code file. Most of the other 8008 mnemonics are done similarly.

Although most of the mnemonics used are standard as compiled in the first versions of Intel 8008 documentation, the INx/OUTxx mnemonics were changed to a Rx/Wxx format to avoid confusion with mnemonics for incrementing registers (e.g., INA).

Some 8008 opcodes are more complicated than those which can be defined using the .DUSR operand. These opcodes fall into the following two categories: all instructions involving labels, and all instructions involving immediate operands. The basic problem is that these instructions have operands which must be evaluated. However these instructions can easily be accommodated

Fig. 1. The INTEL.PS File Listing.

```
        .TITL   INTEL           ; INTEL 8008 PS
        .TXTM   1
        .XPNG                   <-- this deletes old symbols (as used by NOVA)
        .RDX    8
        .RDXO   8
.DUSR   ACA=210
.DUSR   ACB=211
.DUSR   ACC=212
.DUSR   ACD=213
.DUSR   ACE=214
.DUSR   ACH=215
.DUSR   ACL=216
.DUSR   ACM=217
.DUSR   ADA=200
.DUSR   ADB=201
.DUSR   ADC=202
.DUSR   ADD=203
.DUSR   ADE=204
.DUSR   ADH=205
.DUSR   ADL=206
.DUSR   ADM=207
.DUSR   CPA=270
.DUSR   CPB=271
.DUSR   CPC=272
.DUSR   CPD=273
.DUSR   CPE=274
.DUSR   CPH=275
.DUSR   CPL=276
.DUSR   CPM=277
.DUSR   DCB=011
.DUSR   DCC=021
.DUSR   DCD=031
.DUSR   DCE=041
.DUSR   DCH=051
.DUSR   DCL=061
.DUSR   HLT=000
.DUSR   INB=010
.DUSR   INC=020
.DUSR   IND=030
.DUSR   INE=040
.DUSR   INH=050
.DUSR   INL=060
.DUSR   LAB=301
.DUSR   LAC=302
.DUSR   LAD=303
.DUSR   LAE=304
.DUSR   LAH=305
.DUSR   LAL=306
.DUSR   LAM=307
.DUSR   LBA=310
.DUSR   LBC=310
.DUSR   LBD=313
.DUSR   LDE=314
.DUSR   LBH=315
.DUSR   LBL=316
.DUSR   LBM=317
.DUSR   LCA=320
.DUSR   LCB=321
.DUSR   LCD=323
.DUSR   LCE=324
.DUSR   LCH=325
.DUSR   LCL=326
.DUSR   LCM=327
.DUSR   LDA=330
.DUSR   LDB=331
.DUSR   LDC=332
.DUSR   LDE=334
.DUSR   LDH=335
.DUSR   LDL=336
.DUSR   LDM=337
.DUSR   LEA=340
.DUSR   LEB=341
.DUSR   LEC=342
.DUSR   LED=343
.DUSR   LEH=345
.DUSR   LEL=346
.DUSR   LEM=347
.DUSR   LHA=350
.DUSR   LHB=351
.DUSR   LHC=352
.DUSR   LHD=353
.DUSR   LHE=354
.DUSR   LHL=356
.DUSR   LHM=357
.DUSR   LLA=360
.DUSR   LLB=361
.DUSR   LLC=362
.DUSR   LLD=363
.DUSR   LLE=364
.DUSR   LLH=365
.DUSR   LLM=367
.DUSR   LMA=370
.DUSR   LMB=371
.DUSR   LMC=372
.DUSR   LMD=373
.DUSR   LME=374
.DUSR   LMH=375
.DUSR   LML=376
.DUSR   NDA=240
.DUSR   NDB=241
.DUSR   NDC=242
.DUSR   NDD=243
.DUSR   NDE=244
.DUSR   NDH=245
.DUSR   NDL=246
.DUSR   NDM=247
.DUSR   NOP=300
.DUSR   ORA=260
.DUSR   ORB=261
.DUSR   ORC=262
.DUSR   ORD=263
.DUSR   ORE=264
.DUSR   ORH=265
.DUSR   ORL=266
.DUSR   ORM=267
.DUSR   RAL=022
.DUSR   RAR=012
.DUSR   RET=007
.DUSR   RFC=003
.DUSR   RFP=033
.DUSR   RFS=023
.DUSR   RFZ=013
.DUSR   RLC=002
.DUSR   RRC=012
.DUSR   RST0=005
.DUSR   RST1=015
.DUSR   RST2=025
.DUSR   RST3=035
.DUSR   RST4=045
.DUSR   RST5=055
.DUSR   RST6=065
.DUSR   RST7=075
.DUSR   RTC=043
.DUSR   RTP=073
.DUSR   RTS=063
.DUSR   RTZ=053
.DUSR   SBA=230
.DUSR   SBB=231
.DUSR   SBC=232
.DUSR   SBD=233
.DUSR   SBE=234
.DUSR   SBH=235
.DUSR   SBL=236
.DUSR   SBM=237
.DUSR   SUA=220
.DUSR   SUB=221
.DUSR   SUC=222
.DUSR   SUD=223
.DUSR   SUE=224
.DUSR   SUH=225
.DUSR   SUL=226
.DUSR   SUM=227
.DUSR   XRA=250
.DUSR   XRB=251
.DUSR   XRC=252
.DUSR   XRD=253
.DUSR   XRE=254
.DUSR   XRH=255
.DUSR   XRL=256
.DUSR   .DUSR   XRM=257
.DUSR   R0=101
.DUSR   R1=103
.DUSR   R2=105
.DUSR   R3=107
.DUSR   R4=111
.DUSR   R5=113
.DUSR   R6=115
.DUSR   R7=117
.DUSR   W10=121
.DUSR   W11=123
.DUSR   W12=125
.DUSR   W13=127
.DUSR   W14=131
.DUSR   W15=133
.DUSR   W16=135
.DUSR   W17=137
.DUSR   W20=141
.DUSR   W21=143
.DUSR   W22=145
.DUSR   W23=147
.DUSR   W24=151
.DUSR   W25=153
.DUSR   W26=155
.DUSR   W27=157
.DUSR   W30=161
.DUSR   W31=163
.DUSR   W32=165
.DUSR   W33=167

.DUSR   W34=171
.DUSR   W35=173
.DUSR   W36=175
.DUSR   W37=177
.MACRO  CAL
        106
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  CFC
        102
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  CFP
        132
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  CFS
        122
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  CFZ
        112
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  CTC
        142
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  CTP
        172
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  CTS
        162
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  CTZ
        152
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  JFC
        100
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  JFP
        130
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  JFS
        120
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  JFZ
        110
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  JMP
        104
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  JTC
        140
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  JTP
        170
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  JTS
        160
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  JTZ
        150
        (^1)-(((^1)/400)*400)
        (^1)/400
%
.MACRO  ACI
        014
        (^1)-(((^1)/400)*400)
%
.MACRO  ADI
        004
        (^1)-(((^1)/400)*400)
%
.MACRO  CPI
        074
        (^1)-(((^1)/400)*400)
%
.MACRO  LAI
        006
        (^1)-(((^1)/400)*400)
%
.MACRO  LBI
        016
        (^1)-(((^1)/400)*400)
%
.MACRO  LCI
        026
        (^1)-(((^1)/400)*400)
%
.MACRO  LDI
        036
        (^1)-(((^1)/400)*400)
%
.MACRO  LEI
        046
        (^1)-(((^1)/400)*400)
%
.MACRO  LHI
        056
        (^1)-(((^1)/400)*400)
%
.MACRO  LLI
        066
        (^1)-(((^1)/400)*400)
%
.MACRO  LMI
        076
        (^1)-(((^1)/400)*400)
%
.MACRO  NDI
        044
        (^1)-(((^1)/400)*400)
%
.MACRO  SBI
        034
        (^1)-(((^1)/400)*400)
%
.MACRO  SUI
        024
        (^1)-(((^1)/400)*400)
%
        .END
```

Fig. 2. A sample program, WALKL – source input to the assembler.

```
         .TITL   WALKL
         .ENT    START,WALKL
START:   XRA
         LDI     377
         LEI     303
WALKL:   LAE
         RAL
         LEA
         LAD
         RAL
         LDA
         W31
         LAE
         W30
         LCI     300
DELAY:   LAM
         LAM
         DCC
         JFZ     DELAY
         JMP     WALKL
         JMP     .+1
         .END    START
```

by using the assembler's macro facility to perform the evaluations. An example of a macro defined for the CAL opcode is the following:

```
.MACRO  CAL
        106
        (↑1)-(((↑1)
        /400)*400)

        (↑1)/400
```

(Note that the ↑ was printed as a ^ by the line printer.) As an example, when the following line is encountered in the 8008 source code:

CAL S1#$; CALL SOME KRAZY ROUTINE

then the macro will expand this line into the following form:

106
(S1#$)-(((S1#$)/400)*400)
(S1#$)/400

Now, if S1#$ is a label at location 001002 (note that this is a true octal number as opposed to an "Intelese" octal number), the number 001002 is substituted for every occurrence of (S1#$) in the above expansion so that the following expression results:

106
001002-((001002/400)*400)
001002/400

which is evaluated by the assembler using octal integer arithmetic to:

106
002
002

Thus the macro, despite the fact that the assembler was written for a 16-bit machine and keeps track of memory locations in true octal format, is capable of formatting the label's address in the proper Intelese paged format of L(. . .) and H(. . .). The macro facility is used in a similar manner to create an 8-bit octal representation of the operand for an immediate type of opcode. For example, if S1#$ is defined as above, then:

LLI S1#$

is evaluated:

066
002

Implementing the Noval Assembler

The implementation of the assembler with the 8008-defined symbol table is very straightforward. The first step is to type in the source file INTEL.PS as it is listed in Fig. 1. (Note that it is assumed that the user has some familiarity with the Nova test editor and file format.) Then just issue the following command line:

MAC/S/N INTEL.PS

Once this has been done, the assembler can be used with 8008 source files.

Using the Assembler

It is very simple to assemble any source program. The first step is to type the source program into a file using the Nova text editor. An example is shown in Fig. 2. The .TITL statement is used to name the program if desired (the default name is .MAIN) and appears at the top of all output pages from the macro assembler.

The basic format of any statement is:

label: opcode operand ; comment

Both the label and the comment are optional. However, if they are used, the associated punctuation must be included. An operand is only used for branch and immediate statements (e.g., for those opcodes which require the use of the macro facility).

Assembler Output

The assembler output shown in Fig. 3 was obtained by issuing the command line:

MAC WALKL.SR $LPT/L

which assembles the source file WALKL.SR using the 8008 defined symbol table, and provides a listing via the Nova's line printer ($LPT). The leftmost numbers listed give the line number in the source file. The five digit octal number gives the address (or program counter) value. Note that this is not in the Intelese format using page and address within page (although as stated before, label operands are correctly evaluated into Intelese format). The six digit octal number gives the contents of the given address. The leading three zeroes should be ignored. Printed next on the line is the source statement that was assembled into the code shown. In the case of opcodes with operands, the source line is printed, followed by the macro expansion of the source line.

For the Record . . .

Once the user has tried this system, he will no doubt find unique features of the macro assembler which he can utilize to his benefit (see the Data General Corp. *Macro Assembler User's Manual*, order no. 093-000081). However the following features are some of the most useful. The starting address value can be defined by use of the .LOC psuedo-op. Also, the assembler supports repetition and conditional ops such as: .DO, .IFE, .IFG, .IFL, .IFN, .ENDC, .GOTO. These opcodes can allow easier generation of code. Also, there is no reason why further use of the macro facilities can't be used to take care of the double indirect

Fig. 3. The Assembler output for WALKL.

```
 0001 WALKL MACRO REV 02                               14:44:40 07/30/75
01
02
                              .TITL    WALKL
04                             .ENT     START,WALKL
05
06 00000 000250 START:  XRA
07                       LDI      377
08 00001 000036          036
09 00002 000377          (377)-(((377)/400)*400)
10                       LEI      303
11 00003 000046          046
12 00004 000303          (303)-(((303)/400)*400)
13
14 00005 000304 WALKL:  LAE
15 00006 000022          RAL
16 00007 000340          LEA
17 00010 000303          LAD
18 00011 000022          RAL
19 00012 000330          LDA
20 00013 000163          W31
21 00014 000304          LAE
22 00015 000161          W30
23                       LCI      300
24 00016 000026          026
25 00017 000300          (300)-(((300)/400)*400)
26
27 00020 000307 DELAY:  LAM
28 00021 000307          LAM
29 00022 000021          DCC
30                       JFZ      DELAY
31 00023 000110          110
32 00024 000020          (DELAY)-(((DELAY)/400)*400)
33 00025 000000          (DELAY)/400
34                       JMP      WALKL
35 00026 000104          104
36 00027 000005          (WALKL)-(((WALKL)/400)*400)
37 00030 000000          (WALKL)/400
38
39                       JMP      .+1
40 00031 000104          104
41 00032 000033 <- off by 1   (.+1)-(((.+1)/400)*400)
42 00033 000000 <-       (.+1)/400
43              page boundaries not recognized properly
44                        END      START


  0002 WALKL

DELAY 000020          1/27    1/32    1/33
JFZ   000074   MC     1/30
JMP   000101   MC     1/34    1/39
LCI   000156   MC     1/23
LDI   000162   MC     1/07
LEI   000166   MC     1/10
START 000000   EN     1/04    1/06    1/44
WALKL 000005   EN     1/03    1/04    1/14    1/36    1/37
```

addressing problem, etc. Another interesting feature of the macros is that they can be recursive.

Further refinement of this system, currently being considered by the author, is the use of a Fortran post-processor program which could reformat the listing with addresses in Intelese so that they would be easier to read. Also, there is a slight "bug" when using PC relative addressing as was done on line 39 of the program in Fig. 3 (e.g., the JMP .+1 statement). In this case, the low order byte of the jump address is evaluated to one less than it should be. A post-processor could easily identify this case, and correct it. (Until such time, this problem can be avoided by using labels for all branches – which is better programming practice anyway!) Another use of a post-processor could be to punch a paper tape of the object code which could be loaded directly into the 8008 system if a paper tape reader was available.

Disavowal . . .

The fact that the 8008 can execute some opcodes doesn't make it a computer. This was found out the hard way – by building an 8008 system (it seems that while software people claim they could have surmised this from looking at the instruction set, hardware people have to build a working system before they learn . . .). However, there are some uses for an 8008 system. The author is presently working on an article by the title: "How To Make the 8008 Emulate a Computer."

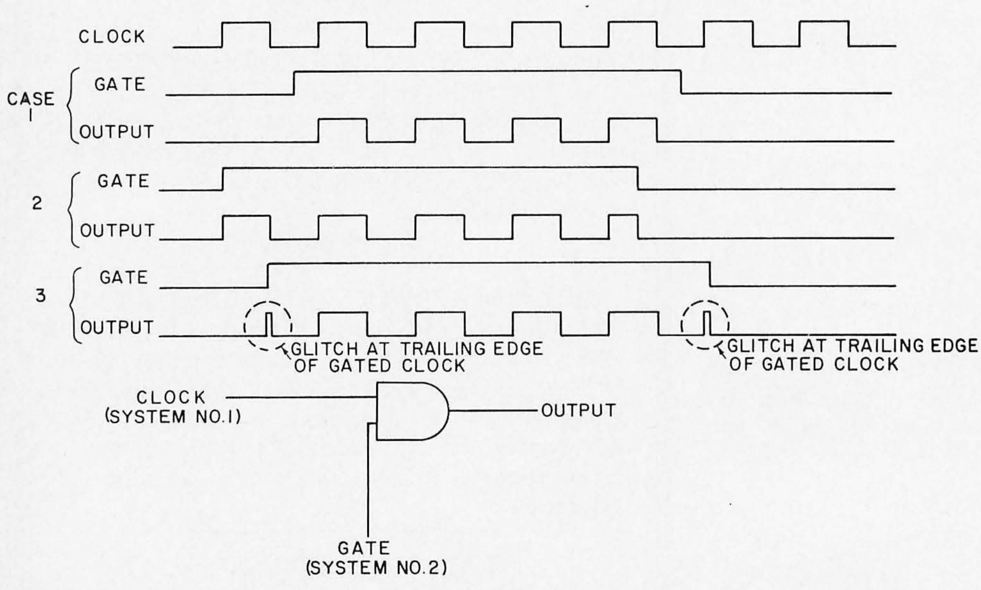
[Look for it in a future issue of BYTE – CH.]

Asynchronitis sufferers – relief is here.

by
Christopher M. Bancroft
CELDAT Design Associates
Box 752
Amherst NH 03031

Asynchronitis

Fig. 1. The Asynchronitis Sufferer. There are cases where the system #1 clock and the system #2 gating signal overlap perfectly (1 and 2) and other pesky cases (like 3) where a glitch develops.



Do you suffer from asynchronitis? A typical sufferer has two systems, each with its own clock, and these systems must communicate. With a simple AND gate their communications suffer, however, because the frequencies are not identical – as we see in Fig. 1, showing Case 1 – effective communication, Case 2 – a bit garbled, and Case 3 – oops! The result in Case 3 shows what happens when the leading edge of the gate and trailing edge of the clock, or vice versa, are *nearly* simultaneous – out comes a short pulse, alias glitch. If counters – or any similar devices – are used, this blip may only partially trigger them, causing erratic or erroneous results.

The chronic sufferer may be cured with a dose constructed from a 7400 package, just a shade more complicated than the single NAND gate. As simple and inexpensive as it is, this dosage guarantees that the gated clock pulses are always complete, eliminating those nasty short spikes which cause bad operation. This cure also provides an output which rises and falls in synchronization with the clock, no matter when the gate command occurs. (See Fig. 2.)

Examining the timing diagram (Fig. 3), we see how the cure works. Gates 1 and 2 (Fig. 2) form a latch that

Fig. 2. How the cure looks. Materials: one 7400 package.

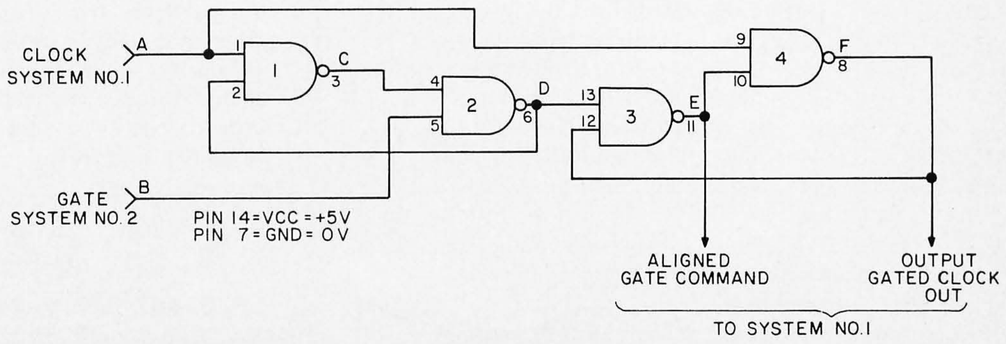
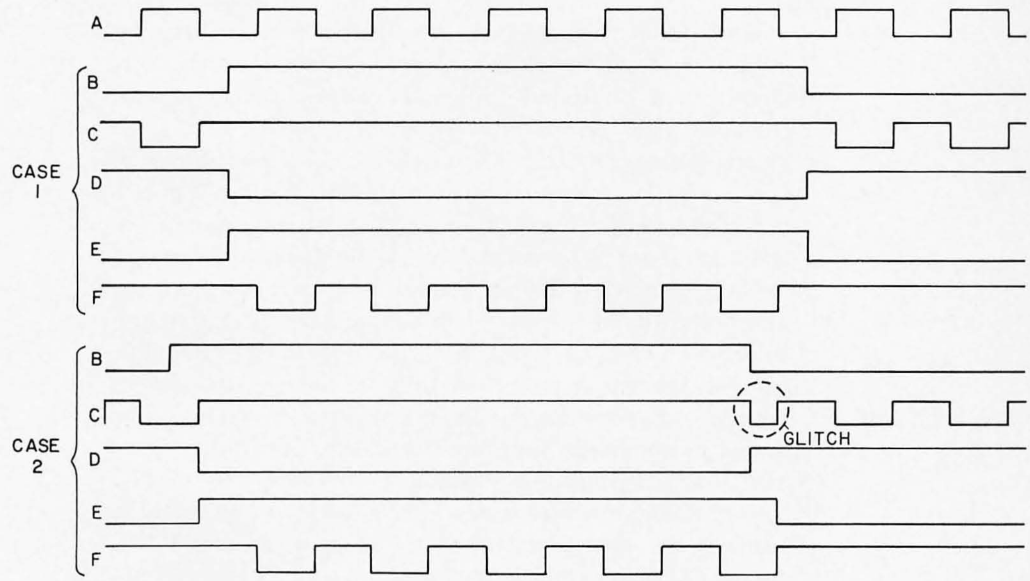


Fig. 3. How the cure works. Case 1 shows the de-glitcher without an active role due to fortuitous alignment. Case 2 shows how a glitch (cropping up at C) is ignored and replaced by a full clock pulse at output F (and a clock synchronized gate level at output E). All gates 7400.



holds the data in the off state when the gate goes high (logic level one) with the clock high. At C, there is a possible glitch out when the gate turns off (Fig. 1, Case 3). However, Gates 3 and 4 form a latch to shield the output from glitches. The output at F, as seen in Fig. 3, is reliably glitch-free. Because E, Fig. 2, is gated in synchronization with the clock, the output at F is synchronous with the clock input.

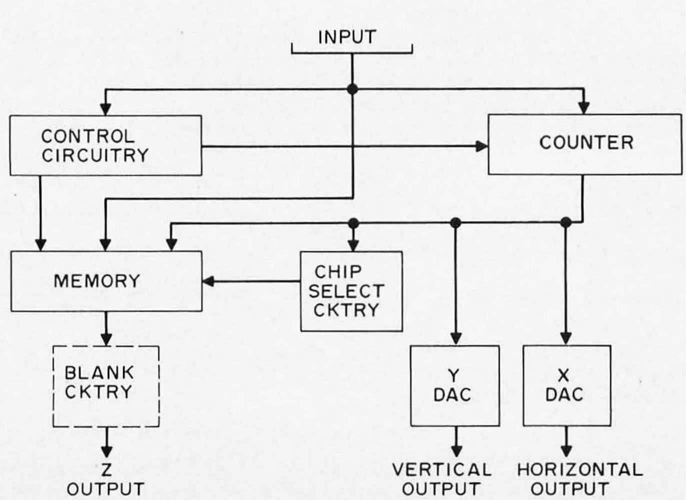
Build an Oscilloscope

Ever wonder how to make a computer draw pictures for output? One way is to use an oscilloscope — which many readers have on general principles for debugging the logic circuitry. Jim Hogenson provides a practical circuit for accomplishing that end in his "Oscilloscope Graphics Interface" design. This graphics device was conceived by Jim as a neat idea to add to the 8008-oriented computer system he was building for a high school science fair. He first mentioned it to me in a letter late last year. I suggested to him (or was it the other way around?) that it might be appropriate to turn it into an article for the *ECS Magazine* I was publishing at the time. After a fair amount of time spent researching the various options — plus one lengthy phone conversation with me — Jim settled on the design shown in this article, which is reprinted here from its original publication in the last issue of *ECS Magazine*. The interface is very simple, and can be adapted to virtually any computer with a minimum of 8 parallel TTL output lines and a clock pulse line which is active when output data is stable. Arrangements have been made for a PC version of this design (see the parts list, Fig. 6) so you won't have to wire wrap the thing like Jim did in his first version.

...CARL

by
James Hogenson
Box 295
Halstad MN 56548

Fig. 1. Oscilloscope graphics display block diagram.



Many members of the large family of alphanumeric computer output devices may be readily used in the home computer system. But there are as yet few devices of a graphic orientation which are economically acceptable in the home computer system. The oscilloscope graphic interface project presented here provides one unique, inexpensive and uncomplicated solution to the graphic output problem in small scale systems. It turns an essential test instrument — the oscilloscope — into a versatile output device.

The oscilloscope graphic interface is programmed and operated through a parallel 8-bit TTL compatible input. An image is represented by a pattern of dots which is organized according to the computer's instructions. During the scan cycle, the digital dot pattern is converted to analog waveforms which reproduce the image on an oscilloscope screen. The graphic interface stores the dot pattern within its own internal refresh memory. Therefore, once the pattern has been generated and loaded into the graphic interface memory, the computer is left free to execute other programs.

Principle of Operation

The raster begins its scan in the upper left-hand corner, scanning left to right and down. The full raster contains 4096 dots, 64 rows of 64 dots each. The horizontal scan is produced by a

Graphics Interface

stepping analog ramp wave. Each of the 64 steps in the ramp produces one dot. The vertical scan is similar. It is a stepping ramp wave consisting of 64 steps. However, there is only one step in the vertical wave for each complete horizontal ramp wave. The result is 64 vertical steps with 64 horizontal steps per vertical step, or 64 rows of 64 dots each.

The timing of horizontal and vertical sweep waveforms originates in a 12-bit binary counter, the operational center of the entire circuit. The six least significant bits of the counter are connected to a digital-to-analog converter (DAC) which converts the digital binary input to a voltage level output. The output of the least significant DAC is the horizontal ramp wave. The six most significant bits are connected to a second DAC. This DAC produces the vertical ramp wave. Incrementing the 12-bit counter at a frequency of around 100 kHz results in a raster on the screen of the oscilloscope.

The contrast in the pattern of dots needed to represent a picture is dependent upon the intensity of each dot. From this point, it is assumed that a dot can be either on or off. An "on" dot will show up on the screen as a bright dot of light. An "off" dot will be a dim dot of light.

When a particular dot is addressed by the counters, it may be set to either the "on" or the "off" state. The on-off control is represented by a single bit. It is this bit which is stored in the internal memory of the oscilloscope graphic interface. There is one bit in the memory for each of the 4096 dots in the raster. When displaying the image, the 12-bit counter which produces the raster addresses the appropriate dot status bit in the memory as that dot is produced on the screen. The on-off dot status bit taken from the memory is converted to a Z-axis signal which controls the intensity of the dot on the screen.

The major portion of the circuitry is taken up in the 12-bit counter, the DACs, and the memory. Fig. 1 shows a block diagram of the oscilloscope graphic interface. The remaining circuitry is the control circuitry which decodes the 8-bit input word and allows for completely programmed operation.

Fig. 2. Oscilloscope graphics interface instruction codes.

Op Code Binary	Octal	Mnemonic	Explanation
00dddddd	0dd	STX	Set X
01dddddd	1dd	STY	Set Y
10xxx000	2x0	DCY	Control – Decrement Y
10xxx001	2x1	TSF	Control – Turn off scan
10xxx010	2x2	ZON	Control – Set Z on
10xxx011	2x3	ZOF	Control – Set Z off
10xxx100	2x4	ZNI	Control – Set Z on with increment
10xxx101	2x5	ZFI	Control – Set Z off with increment
10xxx110	2x6	TSN	Control – Turn on scan
10xxx111	2x7	DCX	Control – Decrement X
11xxxxxx	3xx	CNO	No Op

d = data x = null

Programming

The programming instruction format is shown in Fig. 2. Bits 7 and 6 of the input word are the high-order instruction code. It is assumed that the addressing of dots is done on the basis of X and Y coordinates. The X coordinate is the 6 bits in the least significant or horizontal section of the 12-bit counter. The Y coordinate is the 6 bits in the most significant or vertical section of the counter. In programming from an 8-bit microcomputer source, all 12 bits of the counter cannot be set at once. The counter is set one half or 6 bits at a time. It is for this reason X and Y coordinates are assumed in programming.

When the instruction code (bits 7 and 6) is set at 00, the data on bits 0 through 5 of the input word is loaded into the least significant counter section as the X coordinate. When the instruction code is set at 01, the data on bits 0 through 5 is loaded into the most significant counter section as the Y coordinate. In effect, the Y coordinate will select a row of dots, while the X coordinate will select one dot in the selected row. The coordinates loaded into the counter will address

the memory and select the desired dot status bit for programming.

After loading the coordinates of the dot selected for programming, the status of the dot (on or off) is set using the ZON, ZOF, ZFI or ZNI control codes. Setting the instruction code at 10 directs the control circuitry to decode the three least significant bits of the input word for further instruction. The three least significant bits are called the "control code."

Since the 12-bit counter must store selected coordinates during programming, the raster scan must be disabled before programming. Control code "1" will stop the scan. Control code "6" will restart the scan. When the scan is on, the 12-bit counter will be incremented at a high frequency and the programmed image is displayed on the scope screen.

Control code "2", "set Z on", will program a bright dot to appear at the dot location presently stored in the counter. Control code "3", "set Z off", will program a dim dot or blank to appear at the dot location presently stored in the 12-bit counter.

Control codes "4" and "5" set Z in the same manner as control codes two and three. However, after setting Z, these instructions will increment the counter by one thus advancing to the next dot location in the raster scan pattern. This will allow programming of the entire raster using only a repeated "set Z" instruction.

Control code "0" will not set Z, but will decrement the most significant or vertical section of the counter only. In effect, control code "0" will decrement the stored Y coordinate. Control code "7" will not set Z, but will decrement the entire 12-bit counter by one. This, in effect, will decrement the stored X coordinate. Since the X and Y counter sections are cascaded, Y will automatically be incremented or decremented once for every 64 executions of an increment or decrement X control code.

The increment and decrement control codes are very useful in constructing lines in an image since lines require repeated "set Z" instructions, often on the same axis. An effective method of clearing an image from the screen is repeating a "set Z with increment" control code in a programmed loop. This method allows the option of using either a light or dark image background.

Circuit Operation

Once the data word on the microcomputer parallel output interface is stable, one clock pulse is used to execute the instruction. This clock pulse is taken from the microcomputer output interface. The instruction code is decoded by the 7410 triple three-input NAND gate and two inverters. The clock pulse is enabled by the NAND gate to the appropriate counter section, or to the strobe input of the control code decoder.

The 12-bit counter consists of two 6-bit counting sections. Each section consists of two cascaded TTL 74193 presettable binary counters. Bits 0 through 5 of the data input are common to both sections of the counter. The set X instruction will pulse the load input of the least significant or horizontal section, while the set Y instruction will pulse the load input of the most significant or vertical section of the counter. A pulse on the load input will cause the data on bits 0 through 5 to be loaded into the proper counter section.

Four TTL counters must be used to provide independent loading capabilities for each 6-bit section. The counters within each section are cascaded in the normal fashion. The two sections are cascaded by connecting the upper data B output of the X counter section (IC 8, pin 2) through inverter "a" of IC 2 to the count up input (IC 9, pin 5) of the Y counter section. The inverter is needed to provide proper synchronization in high frequency counting.

The control code is decoded by a 74155 decoder connected for 3 to 8 line decoding. Bits 0 through 2 are decoded by the 74155. The control code is enabled by the pulse coming from the 7410 instruction decoder only when the instruction code is set at 10 on bits 7 and 6.

Decoder lines 1 and 6 are connected to an R/S flip flop

Fig. 3. Timing pulse input to the interface. The 8 data lines must be stable during this pulse.

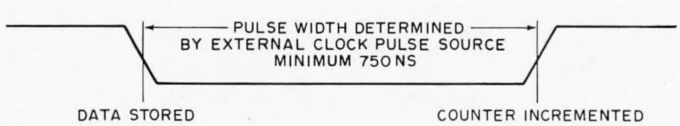


Fig. 4. PC artwork of the graphic interface, by Andrew Hay.
(a) Component side.

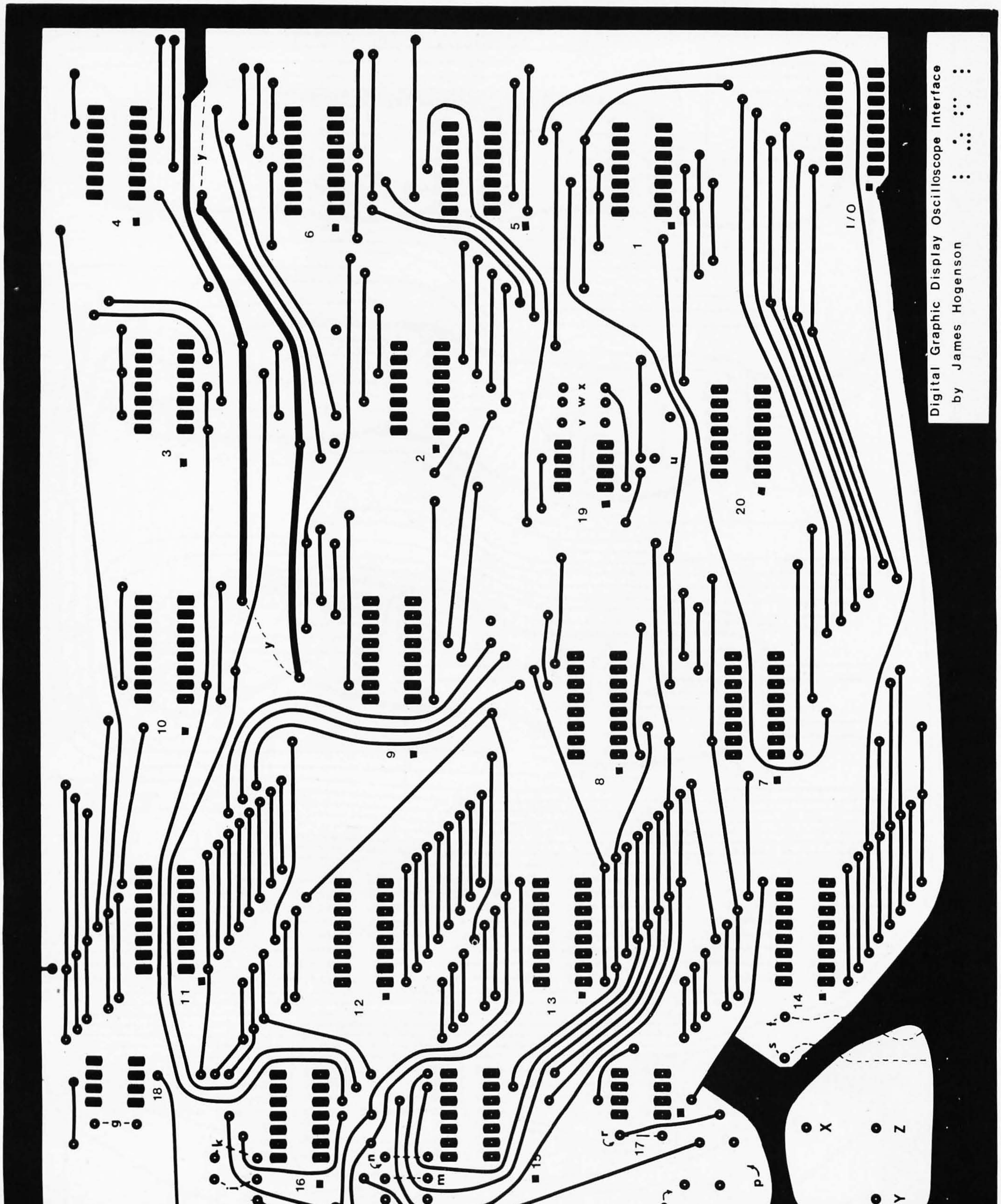
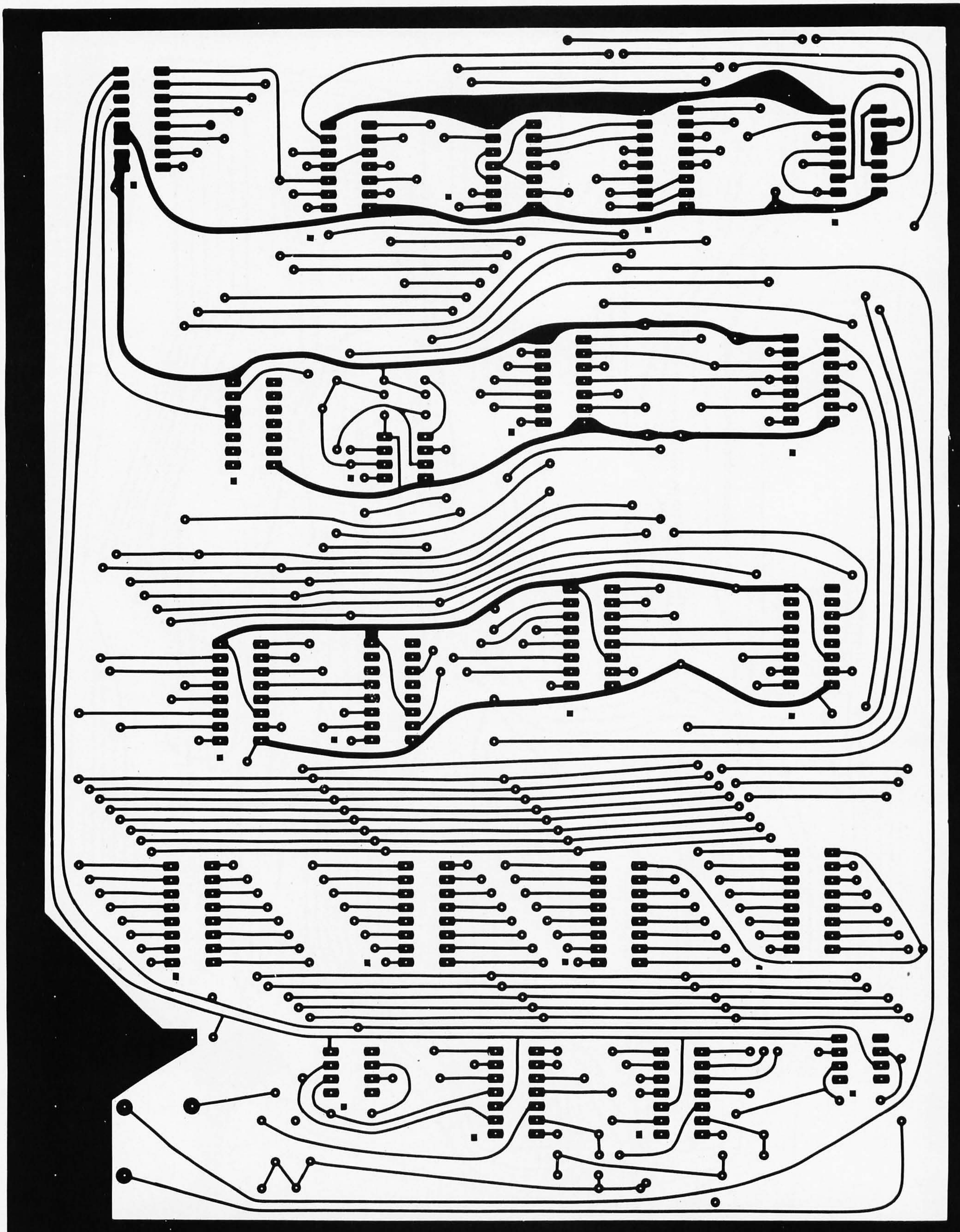


Fig. 4. PC artwork of the graphic interface, by Andrew Hay.
(b) Solder side.



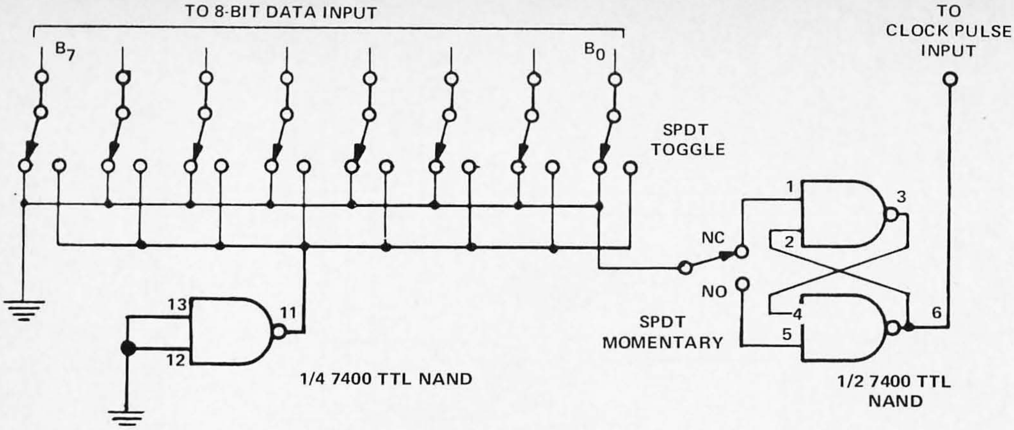


Fig. 5. A test circuit for manual operation. The set-reset flip flop of the 7400 circuit generates a debounced clock pulse which will perform the operation set into the toggle switches. If you haven't got a computer up and running yet, the manual interface can be used in order to test out the display.

which provides the scan on/off control. The flip flop enables the system clock to provide the high frequency square wave which increments the 12-bit counter.

Control codes 2 through 5 define the "set Z" instructions which perform a data write operation. Decoder lines 2, 3, 4 and 5 are connected to a group of AND gates (IC 5a, b, c) functioning as a negative logic OR gate. The output of this gate is the Read/Write control line for the memory. When this line is in the low state, the data present on the data input line of the memory will be written into the memory location presently stored in the 12-bit counter.

The data input of the memory is connected directly to bit 0 of the 8-bit input word. This bit is stored in the memory only when a set Z command is executed. The Z-axis circuit configuration will require a high state pulse for a blank or dim dot. As shown in the binary instruction format, Fig. 2, bit zero will be binary zero for "set Z on" instructions and binary one for "set Z off" instructions. The backward appearance of this binary format will be overlooked when programming in octal notation.

The high frequency system clock controlled by the R/S flip flop and decoder lines 4 and 5 are negative logic ORed. The resulting pulse increments the counter according to control commands.

The same clock pulse taken from the computer output interface is used to write data into the memory and increment the counter in control commands 4 and 5. The data is written into the memory on the leading edge of the pulse. The counter is incremented on the trailing edge. Fig. 3 shows the waveform timing.

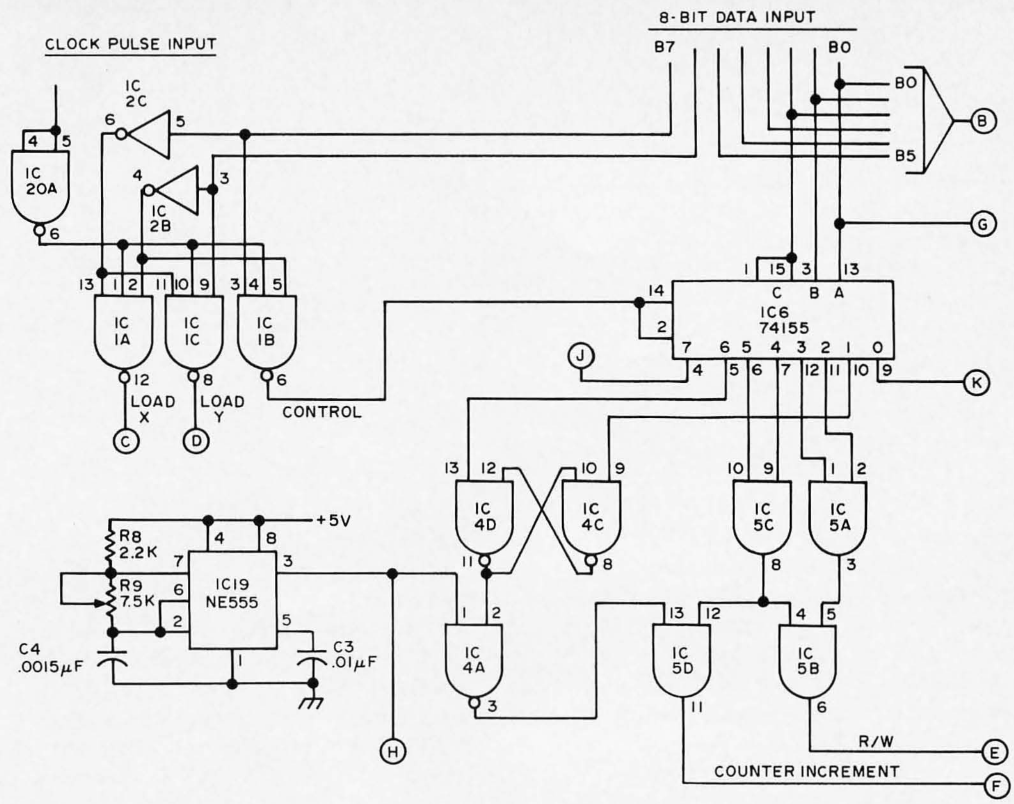
Output bits 0 through 9 of the 12-bit counter are connected to the address inputs of the memory. The memory uses four MM2102 type 1k x 1 bit MOS RAMs (Random Access Memories). Bits 10 and 11 of the counter output are connected to the chip select circuitry which

Fig. 6. Parts list.

C1, C2	20 pF	disc capacitor
C3, C5, C6-C11	.01 mF	disc capacitor
C4	.0015 mF	disc capacitor
C12	25 mF	electrolytic capacitor
IC 1	7410	TTL triple 3-input NAND gate
IC 2	7404	TTL hex inverter
IC 3, IC 4, IC 20	7400	TTL quad 2-input NAND gate
IC 5	7408	TTL quad 2-input AND gate
IC 6	74155	TTL dual 2-to-4-line decoder
IC 7 – IC 10	74193	TTL presettable 4-bit binary counter
IC 11 – IC 14	2102	NMOS 1024-bit static RAM
IC 15, IC 16	MC1406	Motorola 6-bit DAC
IC 17, IC 18	741	Op amp
IC 19	NE555	Oscillator (timer IC)
R1, R2	3.3k Ohm	resistor
R3, R4	5.6k Ohm	resistor
R5, R6	10k Ohm	miniature potentiometer
R7	1k Ohm	resistor (all resistors ¼ Watt, 10%)
R8	2.2k Ohm	resistor
R9	7.5k Ohm	miniature potentiometer

A printed circuit board using the masks of Fig. 4 is available for $29.95. Write to M. F. Bancroft, CELDAT Design Associates, Box 752, Amherst NH 03031.

Fig. 7. Oscilloscope graphics interface circuit diagram. (a)



enables one memory chip at a time for addressing and data input/output operations. The chip select circuitry uses 2 inverters and a TTL 7400 Quad two-input NAND gate.

The data outputs of the RAMs are OR-tied and connected to an AND gate. The data output is synchronized with the high frequency clock for better blanking performance. The output of this gate is connected to the Z-axis blanking circuitry. The blanking circuitry converts the TTL level signal to a scope compatible signal which may be varied over a wide range of output voltages to best match the scope being used.

Bits 0 through 5 of the 12-bit counter are connected to the X coordinate DAC. Bits 6 through 11 are connected to the Y coordinate DAC. The DACs are Motorola MC1406 ICs. The DACs operate on voltages of +5 and -5 to -15. A current output is produced by the DACs. The current output is converted to a voltage output and amplified by the 741 op amps. The output from the X coordinate amp is connected to the horizontal input of the scope. (The scope should be set for external horizontal sweep.) The output from the Y amp is connected to the vertical scope input.

Although the scope used does not need dc-coupled inputs, triggered sweep, or high frequency response for this project, a Z axis or intensity input is required. The Z axis output provided on the interface PC pattern is TTL compatible only. Most scopes will need some type of blanking circuitry to amplify the TTL level pulses. The design of the blanking circuitry will be of the builder's choice, allowing the builder to best suit his scope. A suggested method which is simple and effective is the use of the circuit shown in Fig. 13.

Construction

This project may be wire wrapped, the PC artwork in Fig. 4 may be used to fabricate a double-sided printed circuit board, or the printed circuit board product mentioned in the parts list may be employed. The PC pattern is designed for easy soldering. The components need be soldered on the bottom side only.

Remember that the memory ICs are MOS devices and should be handled as such. Static electricity will easily puncture the thin MOS transistor junctions.

Bypass capacitors should be connected between supply voltages and ground. A minimum of a 10 mF electrolytic or tantalum capacitor should be used for all supply voltages. For the +5 logic supply, one .01 mF disc capacitor should be used for each 2 to 5 integrated circuits. The large

electrolytics will filter out low frequency noise and voltage transients while the small disc capacitors will filter out high frequency noise which could falsely trigger flip flop and counter circuits.

Set-up, Testing and Operation

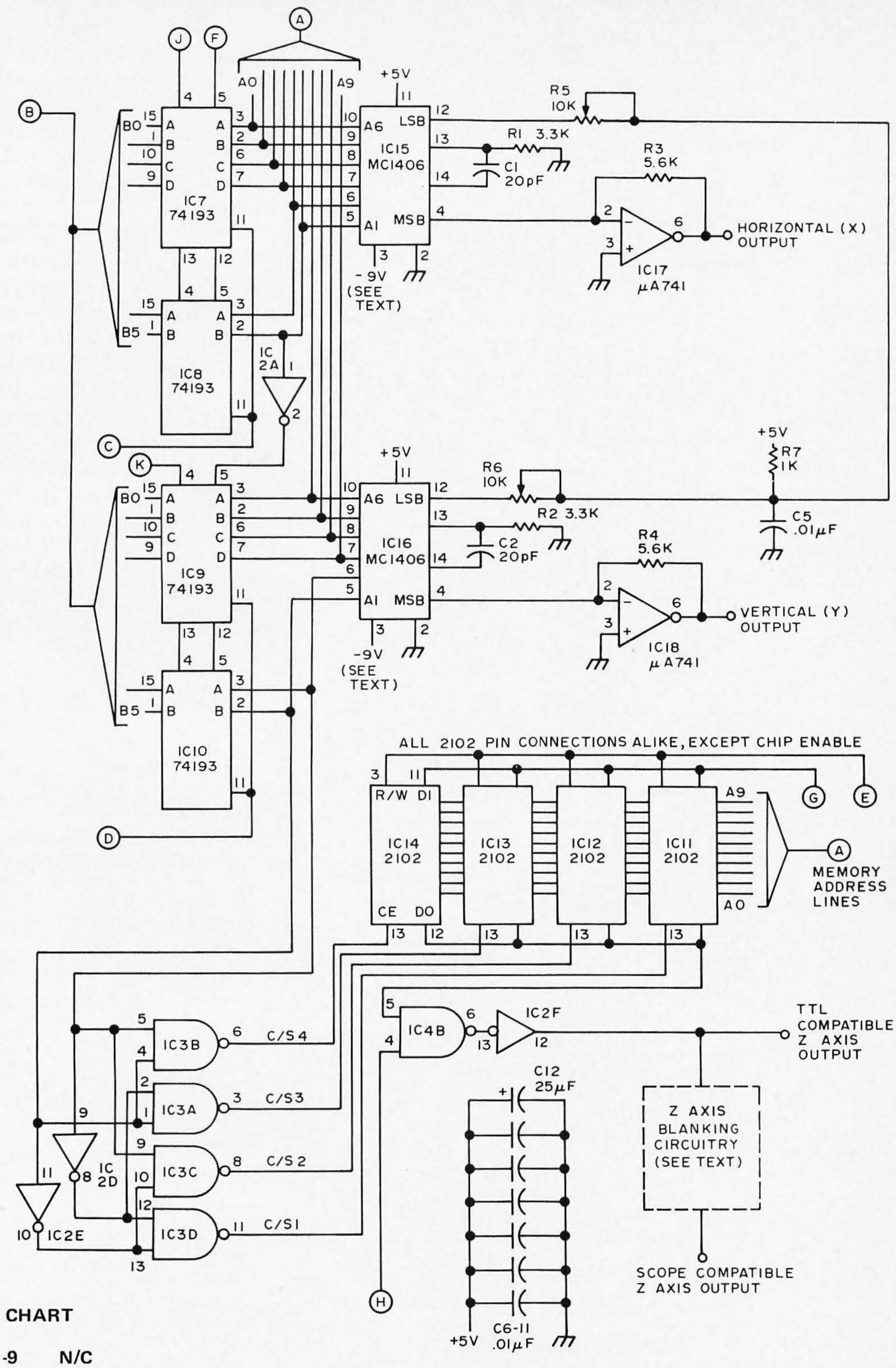
The system requires a +5 volt, 400 mA power supply and a dual polarity supply of from ±9 to ±15 volts at 10 mA. The wide range of analog supply voltages allows use of existing power supplies for the graphic interface.

The clock pulse derived from the computer parallel I/O interface should be active in the low state. If a device operating with an active high pulse is used, one of the free gates of IC 20 may be used to invert the clock pulse or IC 20 may be omitted.

When ready for testing, be certain of voltage supply polarities, then apply power. If the scan does not come on at random, execute a "turn on scan" command. Using the 10k Ohm pots, R5 and R6, adjust the DAC voltage references to eliminate any distorted concentration of dots in the raster.

The system clock consists of a 555 timer IC connected as an astable multivibrator.

Fig. 7. Oscilloscope graphics interface circuit diagram. (b)



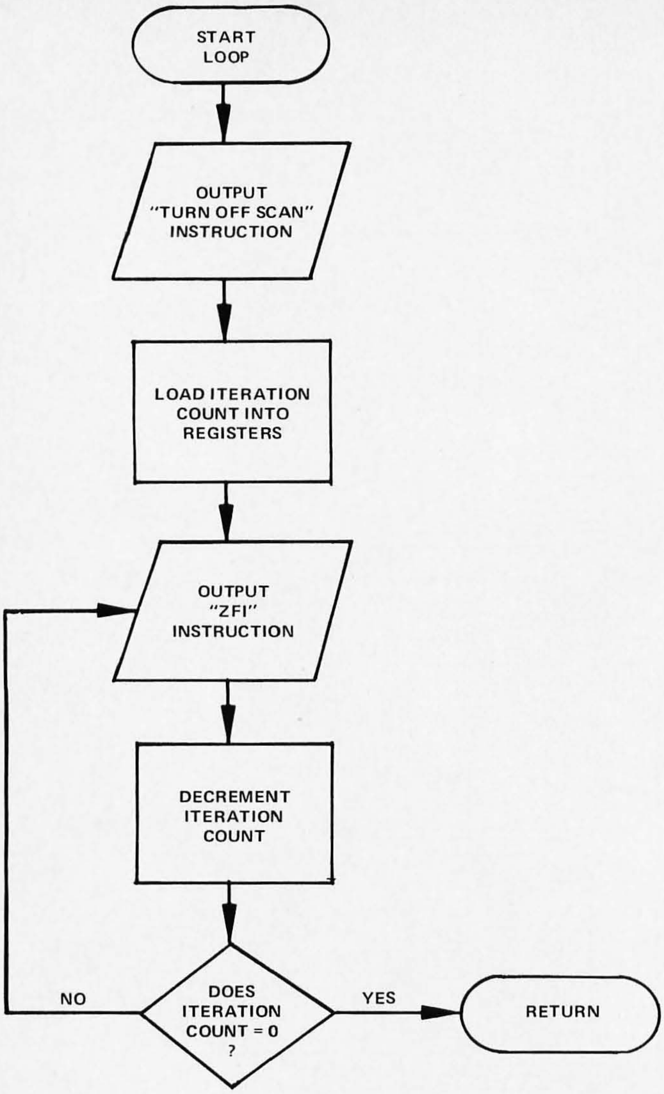
IC POWER AND PIN CONNECTION CHART

IC	+5	GND	+9	-9	N/C
1,2,3,4,5	14	7			
6	16	8			9,4
7, 9	16	8, 14			
8, 10	16	8, 14			6,7,9,10,12,13
11,12,13,14	10	9			
15,16	11	2		3	1
17, 18			7	4	1,5,8
19	4,8	1			
20	14	7			1,2,3,8,9,10,11,12,13

2102 MEMORY ADDRESS PIN CONNECTIONS

A-0 -- pin 8 : A-1 -- pin 4 : A-2 -- pin 5 : A-3 -- pin 6
A-4 -- pin 7 : A-5 -- pin 2 : A-6 -- pin 1 : A-7 -- pin 16
A-8 -- pin 15 : A-9 -- pin 14

Fig. 8. CLEAR Program flow chart.



Adjusting the frequency may be necessary to obtain a stable raster. The frequency is adjusted using R9, the 7.5k pot. The frequency of the system clock should be approximately 100 kHz, but is not critical. The only requirement is appearance of the raster.

If the raster is evenly distributed over the screen, but is severely chopped up, check the digital inputs to the DACs. Use the scope to check the vertical and horizontal ramp waves individually. If the wave is not an even ramp, two or more of the DAC inputs may be reversed. Note that DAC input A1 is the most significant bit while input A6 is the least significant bit. Reversed inputs may also cause incomplete raster formations. Slight gaps or overlapping between some dots is caused by non-linearities in the manufacturing of the DACs.

If no raster at all appears, first check for a square wave output at pin 3 of the 555 timer IC. Then check for square wave outputs at each TTL 74193 counter. These square waves will be binary submultiples of the oscillator frequency. If the counter is operating, check all connections to the DACs and op amps.

Applying power will produce a random pattern of on and off dots. Adjust the amplitude of the Z axis signal for best contrast. Since most scopes will have an ac-coupled (or capacitor coupled) Z axis input, both amplitude and frequency of the signal will affect

Fig. 10. To construct a line segment in the direction shown by the arrow, alternately execute the commands shown.

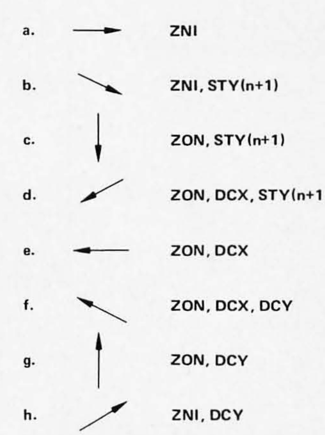


Fig. 9. Listing of 8008 code for the CLEAR program.

START

```
00/344 = 006    LAI
00/345 = 201    (TSF)
00/346 = 121    OUT 10
00/347 = 006    LAI
00/350 = 205
00/351 = 016    LBI
00/352 = 377
00/353 = 026    LCI
00/354 = 021
00/355 = 121    OUT 10
00/356 = 011    DCB
00/357 = 150    JTZ
00/360 = 365
00/361 = 000
00/362 = 104    JMP
00/363 = 355
00/364 = 000
00/365 = 021    DCC
00/366 = 110    JFZ
00/367 = 355
00/370 = 000
00/371 = 377    HLT
```

performance. Charging the capacitor within the scope with too much voltage at a given frequency will cause the blank pulse to carry over into the next dot. This could cause more dots than desired to be blanked out or dimmed.

After a satisfactory raster is obtained, each instruction should be executed to verify its operation. First, clear the screen. The flowchart for a simple CLEAR program is shown in Fig. 8. The method outlined is to simply send out a "set Z off with increment" instruction 4096 times.

Fig. 9 shows the program listing for an 8008 system. This example used the B and C registers to keep track of the iteration count. The register contents are decremented once for each output ZFI instruction. The RETURN instruction may be substituted with a HALT if the CLEAR program is not to be used as a called subroutine. The CLEAR subroutine as listed in Fig. 9 begins by turning off the scan (which must be done before any programming, as stated), but does not turn the scan back on after the interface memory is cleared. The course of operation is left to the programmer once CLEAR has been called.

The chart in Fig. 10 may be used in testing the various control commands. The chart shows the commands to be used to construct a line segment in the direction shown by the arrow. Lines moving in a downward direction require that Y be reset with (n+1) for each dot programmed, "n" being the

Fig. 11. CHECKERBOARD Test Pattern Program flow chart.

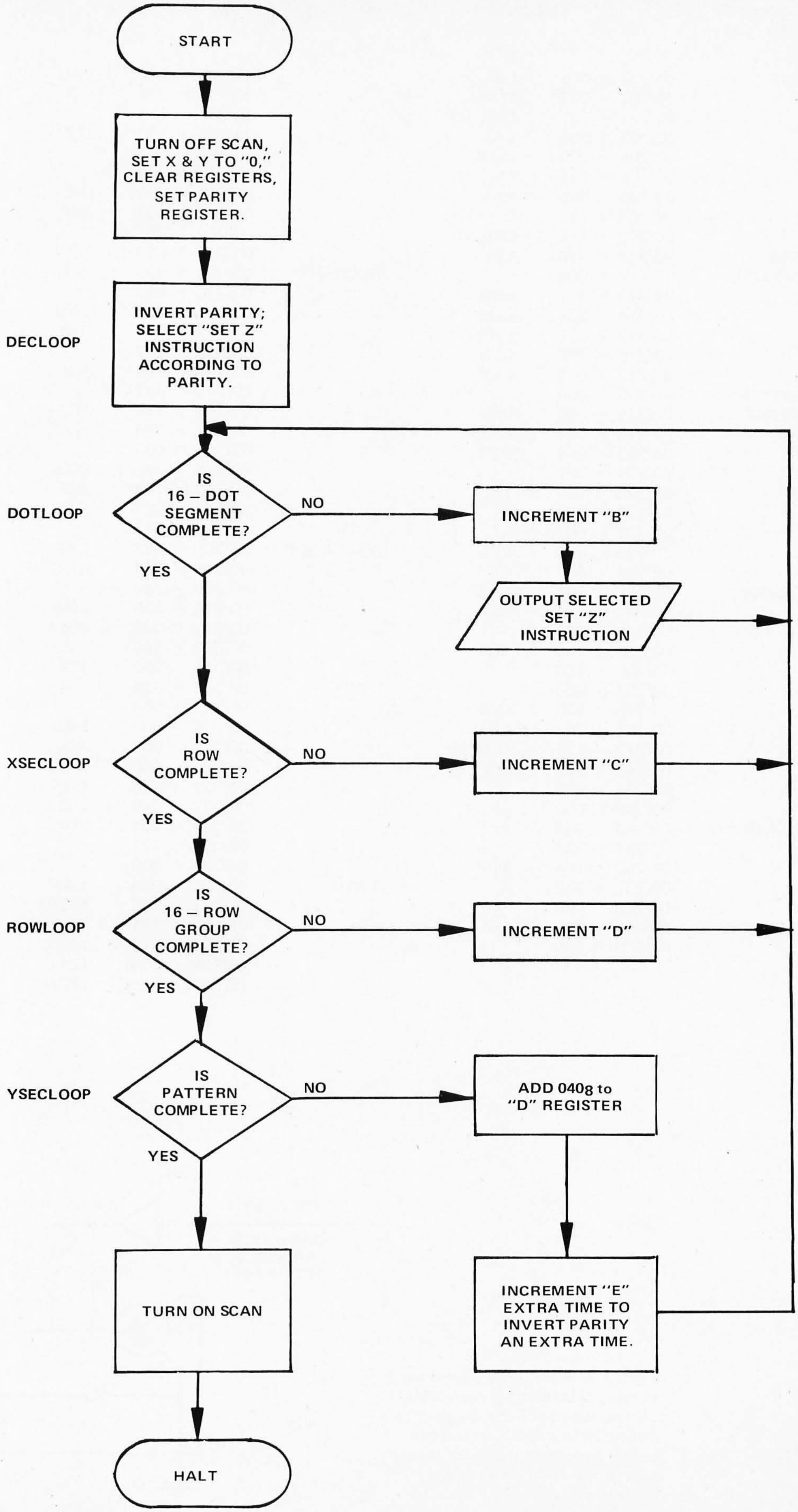


Fig. 12. Listing of 8008 code for the CHECKERBOARD program.

```
START          00/200 = 006   LAI
               00/201 = 201   (TSF)
               00/202 = 121   OUT 10
               00/203 = 006   LAI
               00/204 = 000   (STX)
               00/205 = 121   OUT 10
               00/206 = 006   LAI
               00/207 = 100   (STY)
               00/210 = 121   OUT 10
CLEAR          00/211 = 016   LBI
REGISTERS      00/212 = 000
               00/213 = 321   LCB
               00/214 = 331   LDB
               00/215 = 351   LHB
               00/216 = 361   LLB
               00/217 = 046   LEI
PARITY REG     00/220 = 000
DECLOOP        00/221 = 040   INE
               00/222 = 304   LAE
               00/223 = 044   NDI
               00/224 = 001
               00/225 = 150   JTZ
               00/226 = 246
               00/227 = 000
               00/230 = 066   LLI
               00/231 = 332
DOTLOOP        00/232 = 301   LAB
               00/233 = 024   SUI
               00/234 = 020
               00/235 = 150   JTZ
               00/236 = 253
               00/237 = 000
               00/240 = 010   INB
               00/241 = 307   LAM
               00/242 = 121   OUT 10
               00/243 = 104   JMP
               00/244 = 232
               00/245 = 000
DECLOOPJMP     00/246 = 066   LLI
               00/247 = 333
               00/250 = 104   JMP
               00/251 = 232
               00/252 = 000
XSECLOOP       00/253 = 016   LBI
               00/254 = 000
               00/255 = 302   LAC
               00/256 = 024   SUI
               00/257 = 003
               00/260 = 150   JTZ
               00/261 = 267
               00/262 = 000
               00/263 = 020   INC
               00/264 = 104   JMP
               00/265 = 221
               00/266 = 000
ROWLOOP        00/267 = 026   LCI
               00/270 = 000
               00/271 = 303   LAD
               00/272 = 044   NDI
               00/273 = 037
               00/274 = 024   SUI
               00/275 = 017
               00/276 = 150   JTZ
               00/277 = 305
               00/300 = 000
               00/301 = 030   IND
               00/302 = 104   JMP
               00/303 = 221
               00/304 = 000
YSECLOOP       00/305 = 303   LAD
               00/306 = 044   NDI
               00/307 = 340
               00/310 = 330   LDA
               00/311 = 024   SUI
               00/312 = 140
               00/313 = 150   JTZ
               00/314 = 326
               00/315 = 000
               00/316 = 303   LAD
               00/317 = 004   ADI
               00/320 = 040
               00/321 = 330   LDA
               00/322 = 040   INE
               00/323 = 104   JMP
               00/324 = 221
               00/325 = 000
END            00/326 = 006   LAI
               00/327 = 206   (TSN)
               00/330 = 121   OUT 10
               00/331 = 377   HLT
               00/332 = 204   (ZNI)
               00/333 = 205   (ZFI)
```

present Y coordinate. Use the STX and STY instructions to select a starting point. The dot whose coordinates are X=00, Y=00 will be in the upper left corner, the point where the scan begins its cycle.

The flow chart for a CHECKERBOARD TEST PATTERN program is shown in Fig. 11, with an 8008 listing in Fig. 12. The pattern produced will be 16 alternating light and dark squares. The 64 rows of dots are divided into 4 groups of 16 rows each. Each row is divided into 4 segments. The segments are alternately light and dark. The 4 groups also alternated to reverse the pattern between each group.

The set Z with increment instructions is used. The least significant bit of the E register is used in DECLOOP to alternate between "set Z on" and "set Z off." To obtain the complement of the entire pattern on the screen, place a 001 in location 00/220 instead of 000.

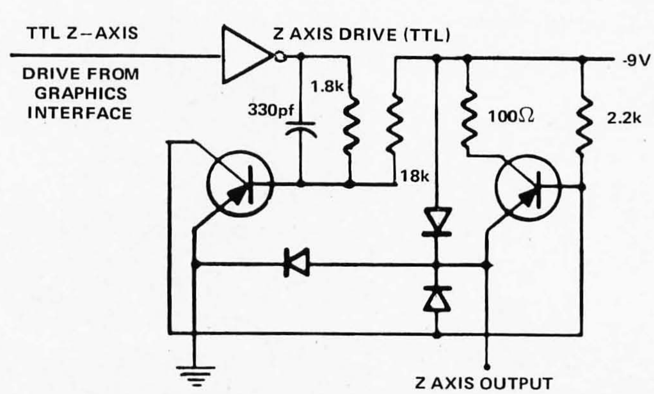
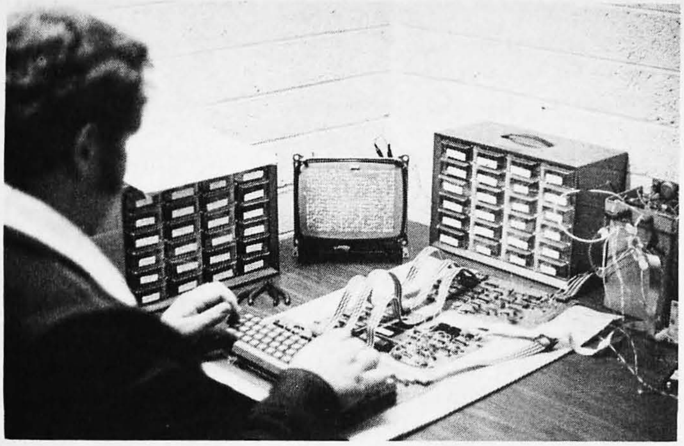


Fig. 13. A Z-axis drive circuit used to control blanking in the author's original version of the design. The transistors are 2N5139s and the diodes are silicon switching diodes such as the 1N914 part or its equivalent.



Mike is demonstrating the prototype Sphere system. It is on three boards, a keyboard/character generator, CPU and memory board. The output is being displayed on a small television screen.



Here's Mike with the Sphere system cabinet and keyboard. In production there will be several more control keys on the keyboard, including a numeric set of keys for fast number entry, cursor control, etc.

from page 61

some of the other 6800 based computer systems which are coming out . . . more on that later.

The plans were to set up a production facility in the back part of the building and have the first kits available in October. This will take some doing, but I wouldn't be surprised if they come close for they are a very determined group.

They have an interesting set of peripherals in the works, including some medium cost floppy disk systems, and a possible revolutionary tape system. We'll try to bring you up to date on Sphere as things progress.

MITS

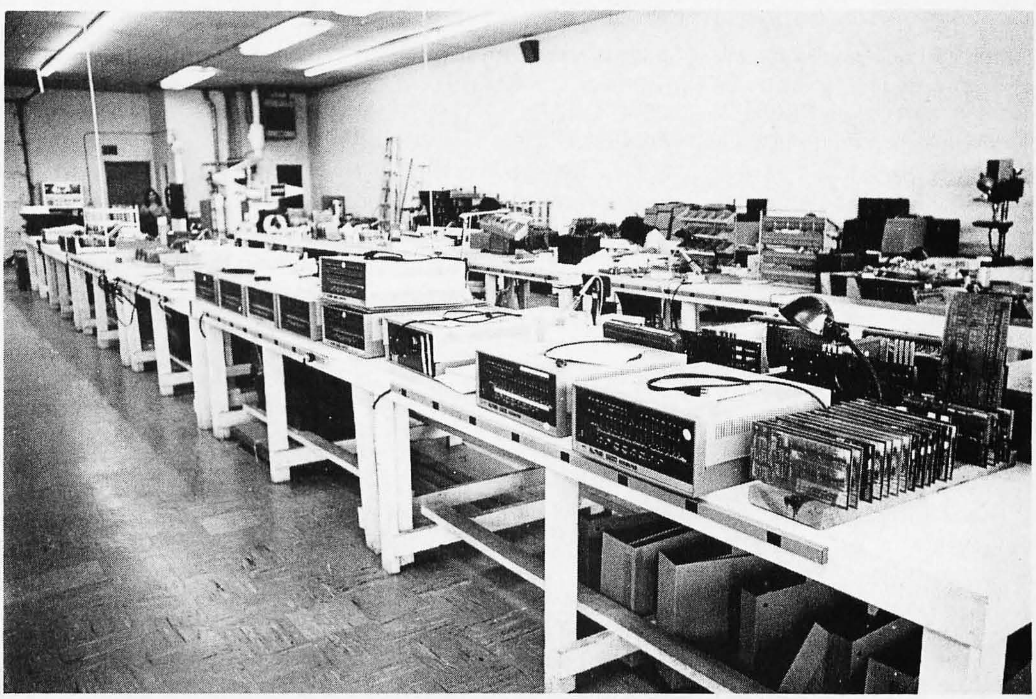
The Altair 8800 has been selling well, as you probably know, and they are busy keeping up with it. They are also in production on all sorts of interface and control boards, memory boards, etc. They also have a very busy group of college-types working away at program development. They are delivering Basic now and are about ready to let loose Extended Basic. They put Basic into a system for me so I could see it work . . . then ran in a tape of a Hammurabi game program and let me sit down and kill off the entire population of a mythical country in short order . . . and become instantly addicted to computer games. I promptly ordered a complete Altair 8800 with enough memory to handle Star Trek. I wanted it right now, but I had to get in line behind the other customers.

Next, in talking with Ed Roberts, the president of MITS, I found that the rumors of a new MITS system based upon the M6800 was much more than a rumor. The dates of release weren't firmed up yet, but it was definitely coming down the pike. Ed said that MITS would give good support to both the Intel 8080 and Motorola M6800 systems. With both Sphere and MITS producing 6800 systems Motorola was doing well and their solid backup of their chips was paying off. This would also mean a big plus to users since this would allow a lot more swapping of programs and would simplify interfacing of memory and peripherals.

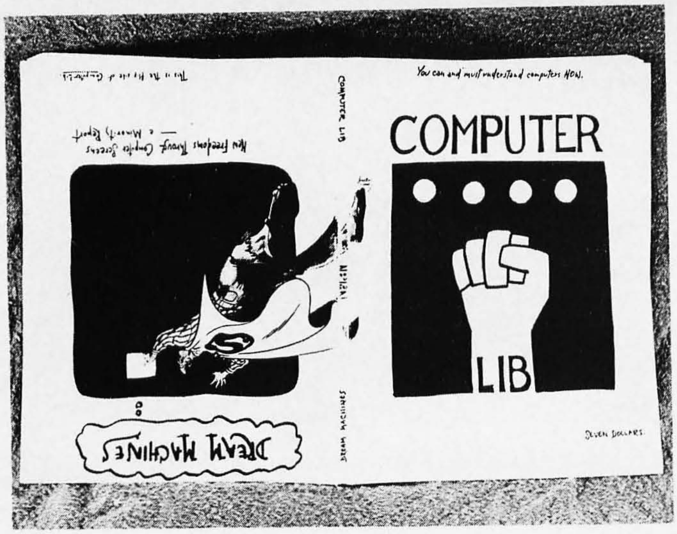
We'll have a lot more info on just what MITS is doing and their plans in the near future.

From New Mexico I flew to San Antonio and a short visit to Southwest Technical Products, Dan Meyer proprietor. SWTP has long been well known for their excellent hi-fi kits. Readers of *The Audio Amateur* (Peterborough NH 03458)

Continued on page 87



Meanwhile, at MITS, over 5000 Altair 8800's have been shipped. Here is a view of part of the production line (during lunch when people were out).





BOOK REVIEW

Computer Lib/Dream Machines *by Theodor H. Nelson. $7 postpaid from Hugo's Book Service, Box 2622, Chicago IL 60690; 10 copies for $50 postpaid.*

This is a marvelous, delightful, one-of-a-kind book. Softbound in an 11" by 14" format and filled with short pieces on many elements of computer lore, this is the *Whole Earth Catalog* of computer fandom. As the author says, a computer fan is "someone who appreciates the options, fun, excitement, and fiendish fascination of computers . . . Somehow the idea is abroad that computer activities are *uncreative,* as compared, say, with rotating clay against your fingers until it becomes a pot. This is categorically false. Computers involve imagination and creation at the highest level. Computers are an involvement you can really get into, regardless of your trip or your karma . . . THEREFORE, welcome to the computer world, the damndest and craziest thing that has ever happened. But we, the computer people, are not crazy. It is you others who are crazy to let us have all this fun and power to ourselves. COMPUTERS BELONG TO ALL MANKIND."

The two halves of the book, *Computer Lib* and *Dream Machines,* start at the two covers and meet at the middle. *Computer Lib* is a general, "cultural" introduction to computers for all those who believe that they will "never understand" what computers are all about. It covers many topics: The basic stored program concept, the "rock bottom" aspects of computer architecture and machine language, minicomputers, big computers, time-sharing, programming languages (with two excellent examples, TRAC* and APL), IBM and the computer industry, "cybercrud," the author's own term for the practice of putting things over on people using computers (especially, forcing them to adapt to a rigid, inflexible, poorly thought out system), and even some comments on "The Hearts and Minds of Computer People."

I found these last comments especially poignant. "Computer people," Nelson says, "are a mystery to others, who see them as somewhat frightening, somewhat ridiculous. Their concerns seem so peculiar, their hours so bizarre, their language so incomprehensible . . . We are like those little people down among the mushrooms, skittering around completely preoccupied with unfathomable concerns and seemingly indifferent to normal humanity. In the moonlight (i.e., pretty late, with snacks around the equipment) you may hear our music." I'll leave some of the more telling comments about computer people for you to read yourself.

The other half of the book, *Dream Machines,* is primarily an introduction to computer graphics, as a way of organizing ideas and expanding one's creative powers – a topic of intense personal interest to the author. Some useful introductory material on display terminals, film output (and computer art), halftone image synthesis, and shading and smoothing objects is included here. The author then describes his far-out ideas for an advanced text-handling system which manipulates "hypertext," and a super graphics system which is the ideal medium for "fantics" (the art of getting ideas across, both emotionally and cognitively) and "thinkertoys." These are then combined into the ultimate system, "Xanadu," which, of course, is part of the "Xanadu network." Besides the material on graphics, this part of the book also features CAI and PLATO, information retrieval, and artificial intelligence.

Nelson is a generalist, for the most part, and, like many generalists, his explanations for things are sometimes overly simplistic. Readers with a strong technical background in certain areas will wince at his explanations of DNA and RNA, the brain, artificial intelligence (which Nelson is, strangely, down on), and "body electronics" (ESB and all that, including "psycho-acoustic dildonics"). And "practically-minded" engineering types, who are involved in the hard work of actually building interactive computer systems, may snicker at Nelson's grandiose plans. But ideas like these are desperately needed, and people who use computers would do well to read this book and share its visions. We're publishing this review in the hope that some of you out there will get the book, seize upon its ideas and turn them into reality. How about it, ladies and gentlemen?

–d.h.f.

***TRAC is a registered service mark of Rockford Research, Inc., which means that we have to print this acknowledgement whenever we so much as mention its name.**

Practical Digital Electronics – An Introductory Course, *by Juris Blukis and Mark Baker. Hewlett-Packard Co., 1501 Page Mill Road, Palo Alto CA 94304. Text (order #05035-90013, $8) and workbook (order # 05035-90003, $8) may be obtained from the company at the above address, or through a local HP sales office.*

This is a good up-to-date, introductory text on digital electronics. It is most notable for its treatment of arithmetic elements and other elementary topics in computer design, data communication codes, and memories.

The text begins with an elementary description of gates and flip flops. Considerable attention is devoted to the simple RS flip flop in order to make sure that the reader understands exactly how this basic memory element works. This is followed by chapters on data communications and logic families. Then counters, shift registers, and other combinational circuits are studied; especially useful are the explanations of priority encoders, which are used to select among interrupts coming from peripheral devices attached to a computer, and parity generator/checkers, which are used in data communications. Finally, there is an excellent chapter on arithmetic elements, and a short chapter on memories (RAMs and ROMs).

While the inclusion of material on memories and data communications is an excellent idea, the actual coverage of these topics is somewhat sparse. One wishes that less space had been devoted to descriptions of the various logic families (27 pages) and more to data communications (7 pages) and memories (13). Not to be missed are some useful appendices on numbering systems and codes, and a brief description of Boolean algebra and Karnaugh maps. Sequential circuits are not introduced in this elementary text.

Whether this book is suitable for you depends on whether you are approaching the topic more from an experimenter's or hobbyist's viewpoint, or from a more "serious" professional engineering viewpoint. If you are a hobbyist, Don Lancaster's *TTL Cookbook* might be more appropriate (see the review in BYTE #1). If you are more "seriously" inclined, this might be the book for you. The accompanying laboratory workbook is independent of the text and is useful only if you are also purchasing HP's 5035T Logic Lab setup.

–d.h.f.



Machine Language Programming For The "8008" (and similar microcomputers) *by Nat Wadsworth. Scelbi Computer Consulting, Inc., 1322 Rear-Boston Post Road, Milford CT 06460. 1975. $19.95.*

Perhaps the best review of *Machine Language Programming For The "8008"* is in the author's own words. "By the time you have completed absorbing and understanding the contents of this publication you should be well equipped to develop programs of your own and thus be in a position to reap even greater benefits from your 8008 based microcomputer . . ."

The book begins with a detailed description of the CPU instruction set and from that point on the reader is led into the intricacies of program development, routines, and sharpening programming skills. Techniques like masking, setting up pointers and counters, and character strings are discussed in detail with illustration by actual source programs. Other chapters cover sorting and mathematical operations, input/output programming, real time programs and PROM considerations. The mathemathical operations chapter gives detailed source programs for floating point arithmetic operations which are directly translatable into machine code for use. This alone is worth the price of the book considering some of the prices quoted for similar programs being offered.

From a different point of view, *Machine Language Programming For The "8008"* is a very unusual book. The author takes a subject matter that has all the possibilities of being dull reading and blends it into a book that is enjoyable to read. One gets the impression that programming is a new art form with the preciseness of science, but the beauty of art. Although the book is based upon the 8008 machines, the material is presented in such a way that it is a worthwhile investment for learning machine language programming.

Bill Fuller
2377 Dalworth 157
Grand Prairie TX 75050

INTEL 8008 TABLE OF OCTAL OP CODES AND "OLD" MNEMONICS

Key: "M" – mnemonic "OP" – op code "L" – length "S" – states "T" – time @ 500 kHz (us).

(Note: the instruction mnemonics used are those of the original 8008 documentation, rather than the later Intel mnemonics for this computer designed for "compatibility"(?) after the fact with the 8080. The early mnemonics are easier to map into op codes using mental gymnastics than the later set.)

M	OP	L	S	T
ACA	210	1	5	20
ACB	211	1	5	20
ACC	212	1	5	20
ACD	213	1	5	20
ACE	214	1	5	20
ACH	215	1	5	20
ACI	014	2	8	32
ACL	216	1	5	20
ACM	217	1	8	32
ADA	200	1	5	20
ADB	201	1	5	20
ADC	202	1	5	20
ADD	203	1	5	20
ADE	204	1	5	20
ADH	205	1	5	20
ADI	004	2	8	32
ADL	206	1	5	20
ADM	207	1	8	32
CAL*	106	3	11	44
CFC	102	3	9/11	36/44
CFP	132	3	9/11	36/44
CFS	122	3	9/11	36/44
CFZ	112	3	9/11	36/44
CPA	270	1	5	20
CPB	271	1	5	20
CPC	272	1	5	20
CPD	273	1	5	20
CPE	274	1	5	20
CPH	275	1	5	20
CPI	074	2	8	32
CPL	276	1	5	20
CPM	277	1	8	32
CTC	142	3	9/11	36/44
CTP	172	3	9/11	36/44
CTS	162	3	9/11	36/44
CTZ	152	3	9/11	36/44
DCB	011	1	5	20
DCC	021	1	5	20
DCD	031	1	5	20
DCE	041	1	5	20
DCH	051	1	5	20
DCL	061	1	5	20
HLT	000	1	X	X
	001	1	X	X
	377	1	X	X
INB	010	1	5	20
INC	020	1	5	20
IND	030	1	5	20
INE	040	1	5	20
INH	050	1	5	20
INL	060	1	5	20

Input – see separate list . . .

M	OP	L	S	T
JFC	100	3	9/11	36/44
JFP	130	3	9/11	36/44
JFS	120	3	9/11	36/44
JFZ	110	3	9/11	36/44
JMP	104*	3	11	44
JTC	140	3	9/11	36/44
JTP	170	3	9/11	36/44
JTS	160	3	9/11	36/44
JTZ	150	3	9/11	36/44
LAB	301	1	5	20
LAC	302	1	5	20
LAD	303	1	5	20
LAE	304	1	5	20
LAH	305	1	5	20
LAI	006	2	8	32
LAL	306	1	5	20
LAM	307	1	8	32
LBA	310	1	5	20
LBC	312	1	5	20
LBD	313	1	5	20
LBE	314	1	5	20
LBH	315	1	5	20
LBI	016	2	8	32
LBL	316	1	5	20
LBM	317	1	8	32
LCA	320	1	5	20
LCB	321	1	5	20
LCD	323	1	5	20
LCE	324	1	5	20
LCH	325	1	5	20
LCI	026	2	8	32
LCL	326	1	5	20
LCM	327	1	8	32
LDA	330	1	5	20
LDB	331	1	5	20
LDC	332	1	5	20
LDE	334	1	5	20
LDH	335	1	5	20
LDI	036	2	8	32
LDL	336	1	5	20
LDM	337	1	8	32
LEA	340	1	5	20
LEB	341	1	5	20
LEC	342	1	5	20
LED	343	1	5	20
LEH	345	1	5	20
LEI	046	2	8	32
LEL	346	1	5	20
LEM	347	1	8	32

Alternatives for conditional instructions: short time if false branch, long time if true branch.

M	OP	L	S	T
LHA	350	1	5	20
LHB	351	1	5	20
LHC	352	1	5	20
LHD	353	1	5	20
LHE	354	1	5	20
LHI	056	2	8	32
LHL	356	1	5	20
LHM	357	1	8	32
LLA	360	1	5	20
LLB	361	1	5	20
LLC	362	1	5	20
LLD	363	1	5	20
LLE	364	1	5	20
LLH	365	1	5	20
LLI	066	2	8	32
LLM	367	1	8	32
LMA	370	1	7	28
LMB	371	1	7	28
LMC	372	1	7	28
LMD	373	1	7	28
LME	374	1	7	28
LMH	375	1	7	28
LMI	076	2	9	36
LML	376	1	7	28
NDA	240	1	5	20
NDB	241	1	5	20
NDC	242	1	5	20
NDD	243	1	5	20
NDE	244	1	5	20
NDH	245	1	5	20
NDI	044	2	8	32
NDL	246	1	5	20
NDM	247	1	8	32
NOP*	300	1	5	20
ORA	260	1	5	20
ORB	261	1	5	20
ORC	262	1	5	20
ORD	263	1	5	20
ORE	264	1	5	20
ORH	265	1	5	20
ORI	064	1	5	20
ORL	266	1	5	20
ORM	267	1	8	32

Output – see separate list . . .

M	OP	L	S	T
RAL	022	1	5	20
RAR	032	1	5	20
RET*	007	1	5	20
RFC	003	1	3/5	12/20
RFP	033	1	3/5	12/20
RFS	023	1	3/5	12/20
RFZ	013	1	3/5	12/20
RLC	002	1	5	20
RRC	012	1	5	20
RST0	005	1	5	20
RST1	015	1	5	20
RST2	025	1	5	20
RST3	035	1	5	20
RST4	045	1	5	20
RST5	055	1	5	20
RST6	065	1	5	20
RST7	075	1	5	20
RTC	043	1	3/5	12/20
RTP	073	1	3/5	12/20
RTS	063	1	3/5	12/20
RTZ	053	1	3/5	12/20
SBA	230	1	5	20
SBB	231	1	5	20
SBC	232	1	5	20
SBD	233	1	5	20
SBE	234	1	5	20
SBH	235	1	5	20
SBI	034	2	8	32
SBL	236	1	5	20
SBM	237	1	8	32
SUA	220	1	5	20
SUB	221	1	5	20
SUC	222	1	5	20
SUD	223	1	5	20
SUE	224	1	5	20
SUH	225	1	5	20
SUI	024	2	8	32
SUL	226	1	5	20
SUM!	227	1	8	32
XRA	250	1	5	20
XRB	251	1	5	20
XRC	252	1	5	20
XRD	253	1	5	20
XRE	254	1	5	20
XRH	255	1	5	20
XRI	054	1	5	20
XRL	256	1	5	20
XRM	257	1	5	20

***Instructions marked with asterisk are typical of several alternate op codes, same function.**

Arithmetic/Logical Mnemonics:

- **AC** = add with carry input
- **AD** = add, no carry input
- **SB** = subtract, borrow input
- **SU** = subtract, no borrow input
- **ND** = logical product (AND)
- **OR** = logical sum (OR)
- **XR** = exclusive or (XOR)
- **CP** = compare

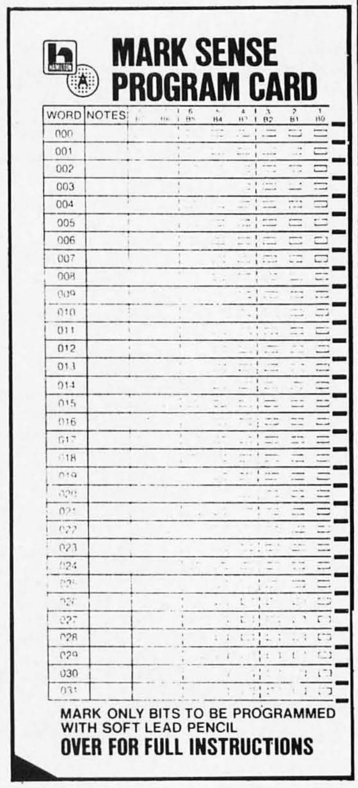
8008 INPUT-OUTPUT LIST

Mnem.	Code	Description*
IN0	101	
IN1	103	
IN2	105	
IN3	107	
IN4	111	
IN5	113	
IN6	115	
IN7	117	
OUT10	121	
OUT11	123	
OUT12	125	
OUT13	127	
OUT14	131	
OUT15	133	
OUT16	135	
OUT17	137	
OUT20	141	
OUT21	143	
OUT22	145	
OUT23	147	
OUT24	151	
OUT25	153	
OUT26	155	
OUT27	157	
OUT30	161	
OUT31	163	
OUT32	165	
OUT33	167	
OUT34	171	
OUT35	173	
OUT36	175	
OUT37	177	

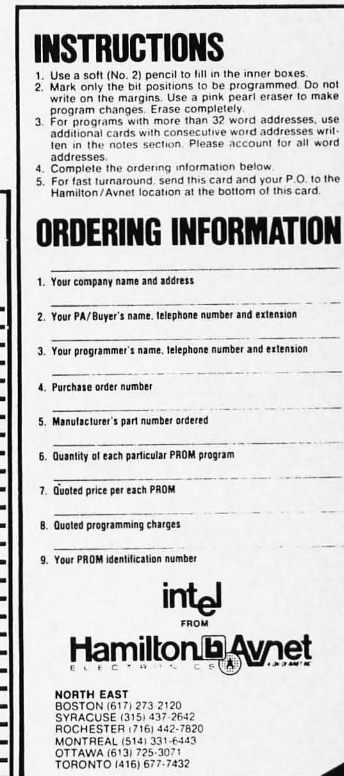
*The description column is left blank for your notes on the current assignments of devices.

Fig. 1. A typical "mark sense card" used to input data to a distributor's programmer.

(a). Programming side – marks are made with a soft lead pencil to indicate which bits are to be "programmed" – to "program" means to change state from the default state to the opposite state. Only those bits which are to be changed (permanently!) are programmed by marks.



(b). Instruction side – notes on use of the card.

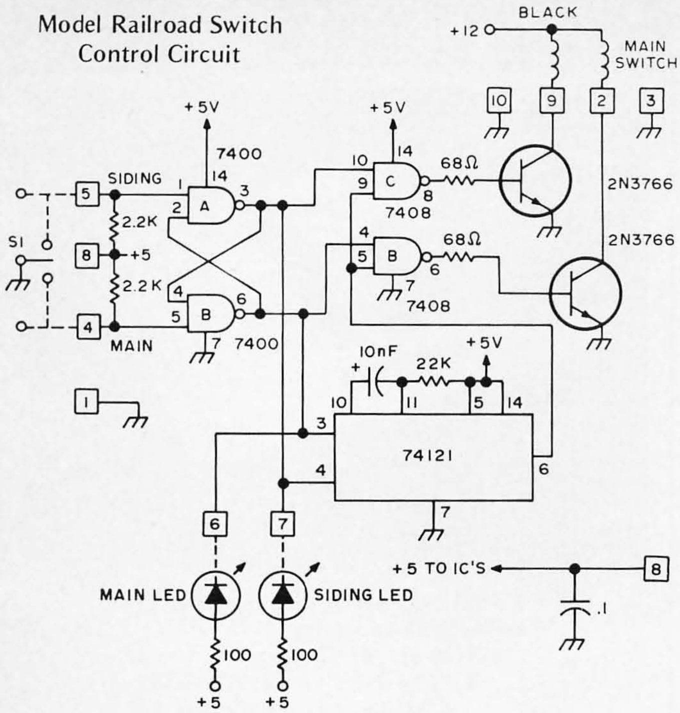


Programming Read-Only-Memories The Easy Way

It is often times advantageous for you to "freeze" a program into one of several types of field programmable read-only memories (PROM is the abbreviation). There are several varieties of read-only memories available, with different characteristics. The purpose of this note is to illustrate one way in which you can get these memories programmed – using the services of an appropriate distributor. Illustrated in Fig. 1 is the method of getting input data to the distributor's programming machines – a "mark sense" card with positions for 32 words of 8 bits. For PROMs of larger sizes, multiple cards are used – see the instructions on the card reproduced in Fig. 1. The only hitch with this method insofar as individuals are concerned is that the distributor's marketing operation is set up to deal with *companies* on a regular basis. Thus it might be best to make an arrangement for one person in a local computer club to handle orders for PROMs by this method – so that the club could be listed as the "company" making the order. The price for programming is nominal – perhaps $2-$3 per chip over the basic cost of the device. (Sooner or later, BYTE will print an article on the various types of field programmable ROM devices and the kind of inexpensive programmers which can be built for home use.)

. . . CARL

Model Railroad Switch Control Circuit



[1], [3], [10] GROUND

[2] TO SWITCH COIL–MAIN } SEPARATE + 12V SUPPLY 1 AMP, (UNREGULATED) USED FOR SWITCHES

[9] TO SWITCH COIL–SIDING

[4] CONTROL–SHORT TO GROUND TO THROW SWITCH TO MAIN LINE

[5] CONTROL–SHORT TO GROUND TO THROW SWITCH TO SIDING

[6] LED TO +5 TO INDICATE SWITCH IN MAIN (THIS POINT LOW)

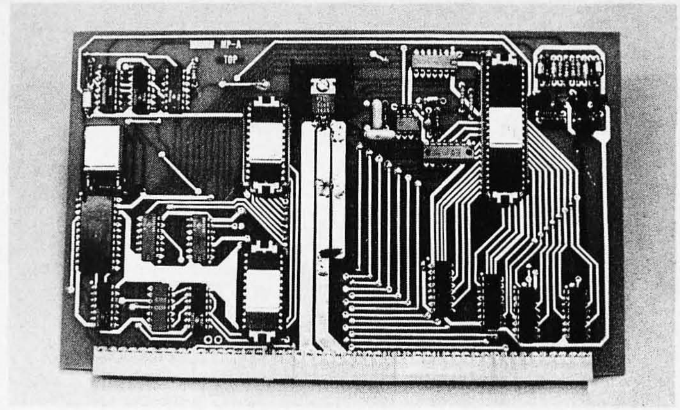
[7] LED TO +5 TO INDICATE SWITCH IN SIDING

[8] +5 VOLTS IN FOR ICs

POINTS 4 AND 5 CAN BE PARALLEL TO MANUAL MOMENTARY SWITCHES AND LOGIC SWITCHES–ANY PULSE (LOW) WILL WORK, HOLDING POWER ON ABOUT ½ SECOND, 74121 WITH RESISTOR AND CAP CONTROL TIME.

Reader Herman De Monstoy is busy working up applications for his 8008 system in the area of model railroad layout control. He sends along this diagram of a model railroad switch control circuit which is used to drive the solenoid operated track switches of a typical HO train layout. The input (at the left of the drawing) can be a pair of complementary TTL signals (e.g., the Q and $\overline{Q}$ outputs of some flip flop) or from the manual switch indicated with dotted lines at the left of the drawing.

The oneshot 74121 is used to control the length of time that the switch is energized in a given direction of movement. The 2N3766 transistors used in this circuit's output drive have a rating of 20 Watts, so this circuit should be able to drive solenoids which take up to about 1 Ampere at 12 volts. Note that it may be necessary to put protection diodes across the coils of the solenoid operated railroad switch if the coil is highly inductive. For manual operation, the electrical push-button switch S1 can be in parallel with computer drive.



And at Southwest Technical Products this prototype CPU board was getting its final wringing out before getting into production. Systems are expected to be available by November! Motorola M6800 based . . . again!

from page 81

have been reading the SWTP ads for some time and seen the rave reviews of the equipment.

Dan has a good sized plant and is doing a substantial business in audio kits. I went there to just say hello and tell him how much I enjoyed putting his television typewriter kit together . . . and to see what he might have up his sleeve for the future. To my amazement he had an M6800 CPU up and going, hooked to one of his TVT units. Those Motorola boys sure do get around. The plans are to have systems available in kit form by November . . . more support for 6800 systems . . . more users, more programs.

This is an exciting time in the microprocessor business with systems getting going just about every month. The Sphere, MITS and SWTP systems are just the vanguard of what is coming. There are outfits talking about some slower systems . . . probably 8008 based . . . which will come in under $200 for the CPU . . . and maybe even one for $100!

All of the firms are working hard to develop accessories, memories and programs. Look out 1976.

PCC Update

In conversation with Bob Albrecht at People's Computer Company, PO Box 310, Menlo Park CA 94025, the following updated information was obtained as BYTE #2 goes to press: PCC's publication will be charging a flat subscription rate of $5 for six issues, with no special rates for students.

FLAKES

It just doesn't take any time at all for the flakes to rise. New as the computer hobby field may be, there are already some sharp operators in there taking advantage of the unwary. I'm put in mind of the "lifetime" guarantee offered by a chap selling ball point pens in the subway cars in New York.

One flake is selling imitation Southwest Tech circuit boards (the television typewriter circuit). Good luck if you fall for this one. SW Tech is a substantial firm with a long history of good products and service to back them up – I've built their TV typewriter and it is splendid. Said flake is getting ready to put out imitation microprocessor boards . . . and kits. Undoubtedly he will give all of the support to his boards and system that one might expect from someone operating out of a cellar.

Watch out!

. . . WAYNE GREEN

5% OFF ON ORDERS OVER $50.00
10% OFF ON ORDERS OVER $100.00
15% OFF ON ORDERS OVER $250.00

TTL

		7451	.17	74154	1.25
7400	$ .14	7453	.17	74155	1.07
7401	.16	7454	.17	74156	1.07
7402	.15	7460	.17	74157	.99
7403	.16	7464	.35	74158	1.79
7404	.19	7465	.35	74160	1.39
7405	.19	7470	.30	74161	1.25
7406	.35	7472	.30	74162	1.49
7407	.35	7473	.35	74163	1.39
7408	.18	7474	.35	74164	1.59
7409	.19	7475	.57	74165	1.59
7410	.16	7476	.39	74166	1.49
7411	.25	7483	.79	74170	2.30
7413	.55	7485	1.10	74173	1.49
7416	.35	7486	.40	74174	1.62
7417	.35	7489	2.48	74175	1.39
7420	.16	7490	.59	74176	.89
7422	.26	7491	.97	74177	.84
7423	.29	7492	.71	74180	.90
7425	.27	7493	.60	74181	2.98
7426	.26	7494	.94	74182	.79
7427	.29	7495	.79	74184	2.29
7430	.20	7496	.79	74185	2.29
7432	.23	74100	1.30	74187	5.95
7437	.35	74105	.44	74190	1.35
7438	.35	74107	.40	74191	1.35
7440	.17	74121	.42	74192	1.25
7441	.98	74122	.45	74193	1.19
7442	.77	74123	.85	74194	1.25
7443	.87	74125	.54	74195	.89
7444	.87	74126	.63	74196	1.25
7445	.89	74141	1.04	74197	.89
7446	.93	74145	1.04	74198	1.79
7447	.89	74150	.97	74199	1.79
7448	1.04	74151	.79	74200	5.90
7450	.17	74153	.99		

LOW POWER TTL

74L00	$ .25	74L51	$ .29	75L90	$1.49
74L02	.25	74L55	.33	74L91	1.45
74L03	.25	74L71	.25	74L93	1.69
74L04	.25	74L72	.39	74L95	1.69
74L06	.25	74L73	.49	74L98	2.79
74L10	.25	74L74	.49	74L164	2.79
74L20	.33	74L78	.79	74L165	2.79
74L30	.33	74L85	1.25		
74L42	1.49	74L86	.69		

HIGH SPEED TTL

74H00	$ .25	74H21	$ .25	74H55	$ .25
74H01	.25	74H22	.25	74H60	.25
74H04	.25	74H30	.25	74H61	.25
74H08	.25	74H40	.25	74H62	.25
74H10	.25	74H50	.25	74H72	.39
74H11	.25	74H52	.25	74H74	.39
74H20	.25	74H53	.25	74H76	.49

8000 SERIES

8091	$ .53	8214	$1.49	8811	$ .59
8092	.53	8220	1.49	8812	.89
8095	1.25	8230	2.19	8822	2.19
8121	.80	8520	1.16	8830	2.19
8123	1.43	8551	1.39	8831	2.19
8130	1.97	8552	2.19	8836	.25
8200	2.33	8554	2.19	8880	1.19
8210	2.79	8810	.69	8263	5.79
				8267	2.59

9000 SERIES

9002	$ .35	9309	$ .79	9601	$ .89
9301	1.03	9312	.79	9602	.79

CMOS

		4016A	.56	4050A	.59
4000A	$ .26	4017A	1.19	4066A	.89
4001A	.25	4020A	1.49	4068A	.44
4002A	.25	4021A	1.39	4069A	.44
4006A	1.35	4022A	1.10	4071A	.26
4007A	.26	4023A	.25	4072A	.35
4008A	1.79	4024A	.89	4073A	.39
4009A	.57	4025A	.25	4075A	.39
4010A	.54	4027A	.59	4078A	.39
4011A	.29	4028A	.98	4081A	.26
4012A	.25	4030A	.44	4082A	.35
4013A	.45	4035A	1.27	4528A	1.60
4014A	1.49	4042A	1.47	4585A	2.10
4015A	1.49	4049A	.59		

74C00	$ .22	74C74	$1.04	74C162	$2.93
74C02	.26	74C76	1.34	74C163	2.66
74C04	.44	74C107	1.13	74C164	2.66
74C08	.68	74C151	2.61	74C173	2.61
74C10	.35	74C154	3.15	74C195	2.66
74C20	.35	74C157	1.76	80C95	1.35
74C42	1.61	74C160	2.48	80C97	1.13
74C73	1.04	74C161	2.93		

OCTOBER SPECIALS

SCHOOL TIME SPECIAL POCKET CALCULATOR KIT

5 function plus constant — addressable memory with individual recall — 8 digit display plus overflow — battery saver — uses standard or rechargeable batteries — all necessary parts in ready to assemble form — instructions included



CALC KIT (WITH BATTERIES) **$12.95**
ASSEMBLED (WITH BATTERIES) **$14.95**
BATTERIES ONLY (DISPOSABLE) SET .... **$ 2.00**

8038 FUNCTION GENERATOR

Voltage controlled oscillator — sine, square, trianglar output 16 PIN DIP **$3.95**

5314 CLOCK CHIP

6 digit multiplexed 7 segment output, fast; slow set, 24 PIN DIP **$3.95**

5739 CALCULATOR CHIPS

9 digit, 4 function, chain operation, 9 V battery operation, 16 sec turn-off 22 PIN DIP **$3.95**

MEMORIES

1103	Fully decoded RAM MOS 1024 bit dynamic 18 PIN DIP **$1.49**
1702A	2048 bit static PROM electrically programmable and erasable, 24 PIN DIP **$15.95**
2102-2	1024 bit N channel static RAM 16 PIN DIP **$3.95**
5261	Fully decoded RAM MOS 1024 bit dynamic 18 PIN DIP **$1.79**

DTL

7402	Quad 2 input NOR gate	$ .13
7420	Dual 4 input NAND gate	.14
7446	BCD to 7 seg driver	.79
7460	Dual 4 input expander	.10
7493	4 bit binary counter	.51
74175	Quad D flip flop	1.25

MEMORIES

1101	256 bit RAM MOS	$ 1.50
1103	1024 bit RAM MOS	3.95
2102	1024 bit static RAM	5.55
5203	2048 bit UV eras PROM	17.95
5260	1024 bit RAM	2.49
5261	1024 bit RAM	2.69
5262	2048 bit RAM	5.95
7489	64 bit ROM TTL	2.48
8223	Programmable ROM	3.69
74200	256 bit RAM tri-state	5.90

CALCULATOR & CLOCK CHIPS

5001	12 DIG 4 funct fix dec	$3.45
5002	Same as 5001 exc btry pwr	3.95
5005	12 DIG 4 funct w/mem	4.95
MM5725	8 DIG 4 funct chain & dec	1.98
MM5736	18 pin 6 DIG 4 funct	4.45
MM5738	8 DIG 5 funct K & mem	5.35
MM5739	9 DIG 4 funct (btry sur)	5.35
MM5311	28 pin BCD 6 dig mux	4.45
MM5312	24 pin 1 pps BCD 4 dig mux	3.95
MM5313	28 pin 1 pps BCD 6 dig mux	4.45
MM5314	24 pin 6 dig mux	4.45
MM5316	40 pin alarm 4 dig	5.39

LED's

MV10B	Red TO 18	$ .22
MV50	Axial leads	.18
MV5020	Jumbo Vis. Red (Red Dome)	.22
	Jumbo Vis. Red (Clear Dome)	.22
ME4	Infra red diff. dome	.54
MAN1	Red 7 seg. .270"	2.19
MAN2	Red alpha num .32"	4.39
MAN4	Red 7 seq. .190"	1.95
MAN5	Green 7 seg. .270"	3.45
MAN6	.6" high solid seq.	4.25
MAN7	Red 7 seq. .270"	1.19
MAN8	Yellow 7 seq. .270"	3.45
MAN66	.6" high spaced seq.	3.75
MCT2	Opto-iso transistor	.61

MULTIPLE DISPLAYS

NSN33	3 digit .12" red led 12 pin fits IC skt.	$1.79
HP5082	5 digit .11 led magn. lens com. cath	3.49
FNA37	9 digit 7 seg led RH dec clr. magn. lens	4.95
SP-425-09	9 digit .25" neon direct interface with MOS/LSI, 180 VDC, 7 seg	1.79

SHIFT REGISTERS

MM5013	1024 bit accum. dynamic	mDIP	$1.75
MM5016	500/512 bit dynamic	mDIP	1.59
SL5-4025	Dual 64 bit static	DIP	1.39

DTL

930	$ .15	937	.15	949	.15
932	.15	944	.15	962	.15
936	.15	946	.15	963	.15

Data sheets on request
With order add $.30 for items less than $1.00 ea.

LINEAR CIRCUITS

300	Pos V Reg (super 723)	TO-5	$ .71
301	Hi Perf Op Amp	mDIP TO-5	.29
302	Volt follower	TO-5	.53
304	Neg V Reg	TO-5	.80
305	Pos V Reg	TO-5	.71
307	Op AMP (super 741)	mDIP TO-5	.26
308	Micro Pwr Op Amp	mDIP TO-5	.89
309K	5V 1A regulator	TO-3	1.35
310	V Follower Op Amp	mDIP	1.07
311	Hi perf V Comp	mDIP TO-5	.95
319	Hi Speed Dual Comp	DIP	1.13
320	Neg Reg 5.2, 12, 15	TO-3	1.19
322	Precision Timer	DIP	1.70
324	Quad Op Amp	DIP	1.52
339	Quad Comparator	DIP	1.58
340K	Pos V reg (5V, 6V, 8V, 12V, 15V, 18V, 24V)	TO-3	1.69
340T	Pos V reg (5V, 6V, 8V, 12V, 15V, 18V, 24V)	TO-220	1.49
372	AF-IF Strip detector	DIP	2.93
373	AM/FM/SSB Strip	DIP	.53
376	Pos V Reg	mDIP	2.42
377	2w Stereo amp	DIP	1.16
380	2w Audio Amp	DIP	1.13
380-8	.6w Audio Amp	mDIP	1.52
381	Lo Noise Dual preamp	DIP	1.52
382	Lo Noise Dual preamp	DIP	.71
550	Prec V Reg	DIP	.89
555	Timer	mDIP	.89
556A	Dual 555 Timer	DIP	1.49
560	Phase Locked Loop	DIP	2.48
562	Phase Locked Loop	DIP	2.48
565	Phase Locked Loop	DIP TO-5	2.38
566	Function Gen	mDIP TO-5	2.25
567	Tone Decoder	mDIP	2.66
709	Operational AMPL	TO-5 or DIP	.26
710	Hi Speed Volt Comp	DIP	.35
711	Dual Difference Compar	DIP	.26
723	V Reg	DIP	.62
739	Dual Hi Perf Op Amp	DIP	1.07
741	Comp Op AMP	mDIP TO-5	.32
747	Dual 741 Op Amp	DIP or TO-5	.71
748	Freq Adj 741	mDIP	.35
1304	FM Mulpx Stereo Demod	DIP	1.07
1307	FM Mulpx Stereo Demod	DIP	.74
1458	Dual Comp Op Amp	mDIP	.62
1800	Stereo multiplexer	DIP	2.48
LH2111	Dual LM 211 V Comp	DIP	1.70
3065	TV-FM Sound System	DIP	.62
3075	FM Det-LMTR & Audio preamp	DIP	.71
3900	Quad Amplifier	DIP	.35
7524	Core Mem Sense AMPL	DIP	.71
8864	9 DIG Led Cath Drvr	DIP	2.25
75150	Dual Line Driver	DIP	1.75
75451	Dual Perepheral Driver	mDIP	.35
75452	Dual Peripheral Driver	mDIP	.35
75453	(351) Dual Periph Driver	mDIP	.35
75491	Quad Seq Driver for LED	DIP	.71
75492	Hex Digit driver	DIP	.80

Satisfaction guaranteed. Shipment will be made via first class mail within 3 days from receipt of order. Add $.50 to cover shipping and handling for orders under $25.00. Minimum order $5.00. California residents add sales tax.

INTERNATIONAL ELECTRONICS UNLIMITED

P.O. BOX 1708/ MONTEREY, CA. 93940 USA
PHONE (408) 659-3171

COMPUTER EXPERIMENTER SUPPLIES

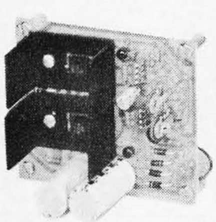
FACTORY FRESH—PRIME QUALITY
PERFORMANCE GUARANTEED

MICROPROCESSORS AND MEMORY

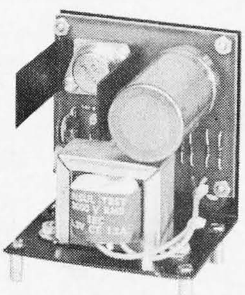
8008	$ 35.00	Commercial Grade—up to 35° C.
8080	135.00	These units are factory fresh, full spec devices.
2102	3.50	
2102-2	4.50	

COMPUTER GRADE REGULATED POWER SUPPLIES

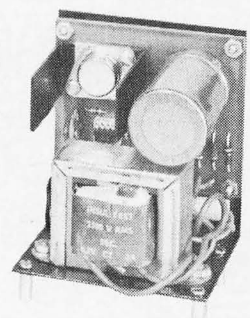
All units are short-circuit proof, fold back current limited and with over-voltage crowbar protection.



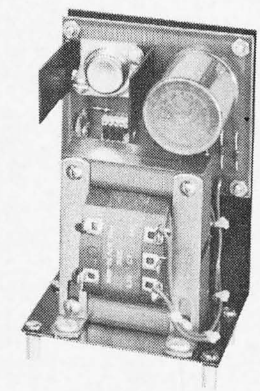
MD-15
±15 Volt at 200MA
Dual Tracking
$30.00



MD-5-1
+5 Volt at 1 Amp
$24.50



MD-5-3
+5 Volt at 3 Amp
$34.50



MD-5-6
+5 Volt at 6 Amp
$44.50

MICRO COMPUTER SUPPLY COMBINATIONS

For the 8008
MD-08—+5 volt at 6 amp, -12, -9 at 200 ma $75.00
For the 8080
MD-80—+5 volt at 6 amp, ±12v at 200 ma ... $75.00
For the Fairchild F-8
MD-8—+5 volt at 6 amp, +12 v at 200 ma ... $65.00
For the M6800
MD-5—+5 volt at 6 amp $44.50

All units are short circuit proof, fold-back current limited and with over voltage crowbar protection.

TTL INTEGRATED CIRCUITS

All devices are factory fresh, full spec units.

7400	.23
7404	.25
7442	.60
7447	.95
7448	.95
7475	.60
7490	.60
7493	.60
74125	.55
74126	.55
74192	1.10
74193	1.10

GUARANTEE

Most devices shipped within 24 hours. If not shippable within 2 weeks payment refunded. Performance guaranteed on all units for 30 days. Defective parts replaced at no charge. NOTICE: This warranty applies **only** to parts that have not been soldered. You must use sockets for your incoming inspection tests.

All Prices Subject to Change Without Notice
Minimum Order $10.00
Add $1.00 to Cover Postage and Handling
Send Check or Money Order (No C.O.D.) To:
N. J. Residents Add 5% Sales Tax

MICRO DIGITAL CORP.

BOX 413, EDISON, NJ 08817 • (201) 549-2699

7400N TTL

SN7400N	.16	SN7451N	.27	SN74151N	1.25
SN7401N	.16	SN7453N	.27	SN74153N	1.35
SN7402N	.21	SN7454N	.41	SN74154N	1.25
SN7403N	.16	SN7459A	.25	SN74155N	1.21
SN7404N	.21	SN7460N	.22	SN74156N	1.30
SN7405N	.24	SN7470N	.45	SN74157N	1.30
SN7406N	.45	SN7472N	.39	SN74160N	1.75
SN7407N	.45	SN7473N	.45	SN74161N	1.45
SN7408N	.25	SN7474N	.45	SN74163N	1.65
SN7409N	.25	SN7475N	.80	SN74164N	1.65
SN7410N	.20	SN7476N	.47	SN74165N	1.65
SN7411N	.30	SN7480N	.50	SN74166N	1.70
SN7412N	.42	SN7482N	1.75	SN74167N	5.50
SN7413N	.85	SN7483N	1.15	SN74170N	3.00
SN7414N	.70	SN7485N	1.12	SN74172N	18.00
SN7416N	.43	SN7486N	.45	SN74173N	1.70
SN7417N	.43	SN7488N	3.50	SN74174N	1.95
SN7418N	.25	SN7489N	3.00	SN74175N	1.95
SN7420N	.21	SN7490N	.59	SN74176N	.90
SN7421N	.39	SN7491N	1.20	SN74177N	.90
SN7423N	.37	SN7492N	.82	SN74180N	1.05
SN7425N	.43	SN7493N	.82	SN74181N	3.55
SN7426N	.31	SN7494N	.91	SN74182N	.95
SN7427N	.37	SN7495N	.91	SN74184N	2.30
SN7429N	.42	SN7496N	.91	SN74185N	2.20
SN7430N	.26	SN74100N	1.25	SN74187N	6.00
SN7432N	.31	SN74107N	.49	SN74190N	1.50
SN7437N	.47	SN74121N	.55	SN74191N	1.50
SN7438N	.40	SN74122N	.49	SN74192N	1.50
SN7439A	.25	SN74123N	1.05	SN74193N	1.40
SN7440N	.21	SN74125N	.60	SN74194N	1.45
SN7441N	1.10	SN74126N	.81	SN74195N	1.00
SN7442N	1.08	SN74132N	3.00	SN74196N	1.25
SN7443N	1.05	SN74141N	1.15	SN74197N	1.00
SN7444N	1.10	SN74142N	6.50	SN74198N	2.25
SN7445N	1.10	SN74143N	7.00	SN74199N	2.25
SN7446N	1.15	SN74144N	7.00	SN74200N	7.00
SN7447N	.89	SN74145N	1.15	SN74251N	2.50
SN7448N	.99	SN74148N	2.50	SN74284N	6.00
SN7450N	.26	SN74150N	1.10	SN74285N	6.00

20% Discount for 100 Combined 7400's

CMOS

CD4000	.25			74C10N	.65
CD4001	.25			74C20N	.65
CD4002	.25	CD4030	.65	74C30N	.65
CD4006	2.50	CD4035	1.85	74C42N	2.15
CD4007	.25	CD4040	2.45	74C73N	1.50
CD4009	.59	CD4042	1.90	74C74	1.15
CD4010	.59	CD4044	1.50	74C90N	3.00
CD4011	.25	CD4046	2.51	74C95N	2.00
CD4012	.25	CD4047	2.75	74C107N	1.25
CD4013	.47	CD4049	.79	74C151	2.90
CD4016	.56	CD4050	.79	74C154	3.00
CD4017	1.35	CD4051	2.98	74C157	2.15
CD4019	.55	CD4053	2.98	74C160	3.25
CD4020	1.49	CD4060	3.25	74C161	3.25
CD4022	1.25	CD4066	1.75	74C163	3.00
CD4023	.25	CD4069	.45	74C164	3.25
CD4024	1.50	CD4071	.45	74C173	2.60
CD4025	.25	CD4081	.45	74C193	2.75
CD4027	.69	74C00N	.39	74C195	2.75
CD4028	1.65	74C02N	.55	80C97	1.50
CD4029	2.90	74C04N	.75		

LINEAR

LM100H	15.00			LM1310N	2.95
LM106H	2.50			LM1351N	1.65
LM171H	3.75	LM373N	3.25	LM1414N	1.75
LM212H	7.00	LM377N	4.00	LM1458C	.65
LM300H	.80	LM380N	1.39	LM1496N	.95
LM301H	3/1.00	LM380CN	1.05	LM1556V	1.85
LM301CN	3/1.00	LM381N	1.79	LM2111N	1.95
LM302H	.75	LM382N	1.79	LM2901N	2.95
LM304H	1.00	NE501K	8.00	LM3065N	.69
LM305H	.95	NE510A	6.00	LM3900N	.55
LM307CN	.35	NE531H	3.00	LM3905N	.60
LM308H	1.00	NE536T	6.00	LM5556N	1.85
LM308CN	1.00	NE540L	6.00	MC5558V	1.00
LM309H	1.10	NE550N	.79	LM7525N	.90
LM309K	1.25	NE553	2.50	LM7528N	2.20
LM310CN	1.15	NE555V	.75	LM7534N	2.20
LM311H	.90	NE565H	1.25	LM7535N	1.25
LM311N	.90	NE565N	1.95	8038B	4.95
LM318CN	1.50	NE566CN	1.95	LM75450	.49
LM319N	1.30	NE567H	1.25	75451CN	.39
LM319D	9.00	NE567V	1.95	75452CN	.39
LM320K-5	1.35	LM703CN	.45	75453CN	.39
LM320K-5.2	1.35	LM709H	.29	75454CN	.39
LM320K-12	1.35	LM709N	.29	75491CN	.79
LM320K-15	1.35	LM710N	.79	75492CN	.89
LM323K-5	10.50	LM711N	.39	75494CN	.89
LM324N	1.80	LM723N	.55	**RCA LINEAR**	
LM339N	1.70	LM723H	.55	CA3013	1.70
LM340K-5	1.95	LM733N	1.00	CA3023	2.15
LM340K-12	1.95	LM739N	1.29	CA3035	2.25
LM340K-15	1.95	LM741CH	3/1.00	CA3039	1.35
LM340K-24	1.95	LM741CN	3/1.00	CA3046	1.15
LM340To-5	1.75	LM741-14N	.39	CA3059	2.46
LM340To-6	1.75	LM747H	.79	CA3060	2.80
LM340To-12	1.75	LM747N	.79	CA3080	.85
LM340To-15	1.75	LM748H	.39	CA3083	1.60
LM340To-24	1.75	LM748N	.39	CA3086	.59
LM350N	1.00	LM1303N	.90	CA3089	3.25
LM351CN	.65	LM1304N	1.19	CA3091	8.25
LM370N	1.15	LM1305N	1.40	CA3123	1.85
LM370H	1.15	LM1307N	.85	CA3600	1.75

DATA HANDBOOKS

7400	Pin-out & Description of 5400/7400 ICS	$2.95
CMOS	Pin-out & Description of 4000 Series ICS	$2.95
LINEAR	Pin-out & Functional Description of Linear Circuits	$2.95

MICROPROCESSOR COMPONENTS

8008	CPU	$24.95	8111	1024 RAM	$ 7.95
8080	CPU	149.95	1702A	2K PROM	19.95
7489	64B RAM	2.95	5203Q	2K PROM	19.95
8599	Tri State 7489	3.50	8223	PROM	3.00
1101	256B RAM	2.25	2401	2K SR	9.95
2102	1K RAM	2.95	2533	1K SSR	7.95
8101	1024 RAM	7.95	AY-5-1013	UART	6.95
91L02-1	1024 RAM	3.95	93410	2568-RAM	3.25

7010	1K	NMOS	RAM	Retains data w/o Power	29.95
2107	4K	RAM	450NS	Access time-22 pin DIP	19.95

91L02 1K Static RAM Direct Replacement for 2102-1 with 40% Less Current Drawn. **3.95**

.Vector General Purpose Logic CARD Board 14.95
*Very High Noise Immunity * Holds 12 ea. 14 pin DIPS
*44 pin Edge Connection

THE KILOBYTE RAM CARD Per Kit 69.95
*Complete 1Kx8 Memory * High Noise Immunity Components
*Single 5v supply *500NS Access Time &Kit includes sockets, ICS & Board

FREE $12.00 Value — With each $75 order of Microprocessor components, get a one year subscription to BYTE the magazine for computer PHREAKS FREE FREE FREE

JAMES FALL SPECIALS

DIODES

IN4001	50V @ 1 Amp	15/$1.00
IN4002	100V @ 1 Amp	15/$1.00
IN4003	200V @ 1 Amp	15/$1.00
IN4004	400V @ 1 Amp	15/$1.00
IN4148	Switching	20/$1.00

TRANSISTORS

2N2907A	PNP Switching	6/$1.00
2N2222A	NPN Switching	6/$1.00
2N3904	NPN Amp	6/$1.00
2N3906	PNP Amp	6/$1.00
2N918	NPN RF	6/$1.00
2N5951	NJ Fet	6/$1.00
C106B1	3.6 Amp SCR	2/$1.00

TTL/LINEAR

7400	Gate	7/$1.00
7447	Decoder	.79
7490	Counter	.49
74100	8 Bit Latch	.99
74154	Decoder	.99
74197	Counter	.75
LM301H—LM741H		4/$1.00
LM324N—Quad 741-		1.19
LM309H	5V Reg TO-5	.79
LM309K	5V Reg TO-3	.99

DYNAMIC SHIFT REGISTERS

MM500H MM504H MM510H 2 FOR
MM503H MM506H-MM5016H $1.00

CLOCK & CALC. CHIPS

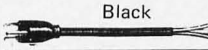
MM5311N	6 Digit	$3.95
MM5312N	4-Digit	3.95
MM5313N	6 Digit	3.95
MM5314N	6 Digit	3.95
MM5316N	6 Digit Alarm	4.95
MM5725N	8 Digit-4 Funt.	1.98
MM5736N	6 Digit-4 Funt.	3.95
MM5738N	8 Digit-5 Funt.	4.95

26 Awg RIBBON CABLE

1 Ft. Minimum	1-9 ft.	10 ft.
4 Cond.	.49 ft.	.39 ft.
8 Cond.	.89 ft.	.69 ft.
16 Cond.	1.49 ft.	1.19 ft.
32 Cond.	2.49 ft.	1.99 ft.

4' POWER SUPPLY CORDS . . . .

Black SPECIAL 3/$1.00



POCKET CALCULATOR KIT

5 function plus constant — addressable memory with individual recall — 8 digit display plus overflow — battery saver — uses standard or rechargeable batteries — all necessary parts in ready to assemble form — instructions included. 3" x 5¼" **SPECIAL $12.95 each**



OPTIONS —
115VAC Transformer . . . . 4.95 each
6 each "N" Alkaline Batteries . . . . . 1.95 lot

THUMBWHEEL SWITCHES

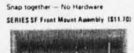


THUMBWHEEL SWITCH ONLY
SF — Front Mount SR — Rear Mount

Part No.	Description	Price
SF-12	Single Pole 10 Position	$2.50
SR-12	Decimal	3.00
SF-21	10 Position BCD only	2.50
SR-21		3.00

Snap together — No Hardware

SERIES SF Front Mount Assembly ($11.70)



Note: Picture to the left has the following 4 ea. SF-12, 1 ea. SF-EP, 1 ea. SF-BB, 1 ea. SF-DP and 1 ea. SF-HB

ACCESSORIES

Part No.	Description	Price
SF-EP	End Plate (Pair)	.50
SF-DP	Divider Plate (each)	.40
SF-BB	Blank Body (each)	.40
SF-HB	Half Body (each)	.40

SERIES SR Rear Mount Assembly ($13.70)



Note: Picture to the left has the following 4 ea. SR-12, 1 ea. SR-EP, 1 ea. SR-BB, 1 ea. SR-DP and 1 ea. SR-HB

ACCESSORIES

Part No.	Description	Price
SR-EP	End Plates (pair)	$ .50
SR-DP	Divider Plate (each)	.40
SR-BB	Blank Body (each)	.40
SR-HB	Half Body (each)	.40

Ordering: Order desired switch or switches and add necessary accessories for your particular application.

.394" DIAM. TRIMMER

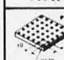


STANDARD RESISTANCE VALUES
MODEL 100Ω 500Ω 1K 2K 5K 10K 20K 100K 200K 1Meg

SPECIAL *Model TR 11*

RESISTANCES (OHMS)	1-9	10-24
STD 100Ω 1 MEGΩ	4/$1.00	

1/16 VECTOR BOARD



P-Pattern
0.1" Hole Spacing

MATERIAL	STOCK NO.	LONG	WIDE	PRICES 1	20
PHENOLIC	64P44 062XXXP	4.50	6.50	1.72	1.54
	169P44 02XXXP	4.50	17.00	3.69	3.32
EPOXY GLASS	64P44 062	4.50	6.50	2.07	1.86
	84P44 062	4.50	8.50	2.56	2.31
	169P44 062	4.50	17.00	5.04	4.53
	169P84 062	8.50	17.00	9.23	8.26
EPOXY GLASS COPPER CLAD	169P44 062C1	4.50	17.00	6.80	6.12

NEW PROTO BOARD-100

Here's a low cost, big 10 IC capacity breadboard kit with all the quality of QT Sockets and the best of the Proto Board series . . . complete down to the last nut, bolt and screw. Includes 2 QT-35S Sockets; 1 QT-35B Bus Strip; 2 5-way binding posts; 4 rubber feet; screws, nuts, bolts; and easy assembly instructions.



COMPLETE KIT . . . *Special **$17.95***

WALL or T.V. DIGITAL CLOCK

12 or 24 Hour
115 Vac
25' VIEWING DISTANCE
Walnut Case-6" x 3" x 1"
Hr. & Min.-6" High
Seconds-3" High



KIT - All Comp. & Case **$34.95**
Wired & Assembled **$39.95**

DISPLAY LEDS

FND70	Com. Cath.	.250	.69
MAN 1	Com. Ano.	.270	$1.95
MAN 2	5x7 Matrix	.300	3.95
MAN 3	Com. Cath.	.125	.39
MAN 4	Com. Cath.	.187	1.95
MAN 7	Com. Ano.	.30	1.50
DL33	Com. Cath.	.125	1.95
DL747	Com. Ano.	.625	2.50

DISCRETE LEDS

MV 10	Red	5/$1.00
MV 50	Red	6/$1.00
MV 5024	Red	5/$1.00
MV 5024	Green	4/$1.00
MV 5024	Yellow	4/$1.00
MV 5024	Orange	4/$1.00
MV 50*	Red 1¼" Leads	5/$1.00



IC SOLDERTAIL — LOW PROFILE (TIN) SOCKETS

	1-24	25-49	50-100		1-24	25-49	50-100
8 pin	$.17	.16	.15	24 pin	$ .38	.37	.36
14 pin	.20	.19	.18	28 pin	.45	.44	.43
16 pin	.22	.23	.22	36 pin	.60	.59	.58
18 pin	.29	.28	.27	40 pin	.63	.62	.61
22 pin	.36	.37	.36				

SOLDERTAIL STANDARD (TIN)

	1-24	25-49	50-100		1-24	25-49	50-100
14 pin	$.30	.28	.26	28 pin	$ .99	.90	.81
16 pin	.33	.31	.29	36 pin	1.39	1.26	1.15
18 pin	.42	.39	.37	40 pin	1.59	1.45	1.30
24 pin	.59	.54	.49				

SOLDERTAIL STANDARD (GOLD)

	1-24	25-49	50-100		1-24	25-49	50-100
8 pin	$.30	.27	.24	24 pin	$ .70	.63	.57
14 pin	.35	.32	.29	28 pin	1.10	1.00	.90
16 pin	.38	.35	.32	36 pin	1.55	1.40	1.26
18 pin	.52	.47	.43	40 pin	1.75	1.59	1.45

WIRE WRAP SOCKETS (GOLD) LEVEL #3

	1-24	25-49	50-100		1-24	25-49	50-100
10 pin	$.45	.41	.37	24 pin	$1.05	.95	.85
14 pin	.39	.38	.37	28 pin	1.40	1.25	1.10
16 pin	.43	.42	.41	36 pin	1.59	1.45	1.30
18 pin	.75	.68	.62	40 pin	1.75	1.55	1.40

50 PCS. RESISTOR ASSORTMENTS $1.75 PER ASST.

ASST. 1	5 ea:	10 OHM / 27 OHM	12 OHM / 33 OHM	15 OHM / 39 OHM	18 OHM / 47 OHM	22 OHM / 56 OHM	1/4 WATT 5% = 50 PCS.
ASST. 2	5 ea:	68 OHM / 180 OHM	82 OHM / 220 OHM	100 OHM / 270 OHM	120 OHM / 330 OHM	150 OHM / 390 OHM	1/4 WATT 5% = 50 PCS.
ASST. 3	5 ea:	470 OHM / 1.2K	560 OHM / 1.5K	680 OHM / 1.8K	820 OHM / 2.2K	1K / 2.7K	1/4 WATT 5% = 50 PCS.
ASST. 4	5 ea:	3.3K / 8.2K	3.9K / 10K	4.7K / 12K	5.6K / 15K	6.8K / 18K	1/4 WATT 5% = 50 PCS.
ASST. 5	5 ea:	22K / 56K	27K / 68K	33K / 82K	39K / 100K	47K / 120K	1/4 WATT 5% = 50 PCS.
ASST. 6	5 ea:	150K / 390K	180K / 470K	220K / 560K	270K / 680K	330K / 820K	1/4 WATT 5% = 50 PCS.
ASST. 7	5 ea:	1M / 2.7M	1.2M / 3.3M	1.5M / 3.9M	1.8M / 4.7M	2.2M / 5.6M	1/4 WATT 5% = 50 PCS.

PRIME INTEGRATED CIRCUIT ASSORTMENTS

ASST. 8	3 ea.	SN7400 / SN7410	7401 / 7430	7402 / 7438	7403 / 7440	7404 / 7472	SSI/TTL	$5.95 ASST.
ASST. 9	2 ea.	SN7447 / SN74145	7490 / 74175	7491 / 74180	74100 / 74191	74121 / 74193	MSI/TTL	$9.95 ASST.
ASST. 10	2 ea.	CD4001 / CD4016	4002 / 4017	4011 / 4019	4012 / 4023	4013 / 4030	CMOS	$7.95 ASST.
ASST. 11	2 ea.	LM301T / LM311T	301N / 565T	302T / 567T	307T / 741T	309K / 741N	LINEAR	$10.95 ASST.

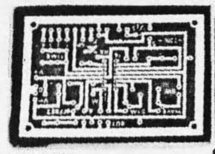
Satisfaction Guaranteed. $5.00 Min. Order. U.S. Funds.
California Residents — Add 6% Sales Tax
Write for FREE 1975F Catalog — Data Sheets .25¢ each

JAMES

P.O. BOX 822, BELMONT, CA. 94002
PHONE ORDERS — (415) 592-8097

IC'S EXAR KITS

FUNCTION GENERATOR KIT



features sine, triangle and square wave; THD 0.5% typ.; AM/FM capability

XR-2206KA SPECIAL $17.95
Includes monolithic function generator IC, PC board, and assembly instruction manual.

XR-2206KB SPECIAL $27.95
Same as XR-2206KA above and includes external components for PC board.

TIMERS		
XR-555CP	Monolithic Timer	SPECIAL $ .69
XR-320P	Precision Timer	1.55
XR-556CP	Dual-555 Timer	1.85
XR-2556CP	Dual Timing Circuit	3.20
XR-2240CP	Programmable Counter/Timer	SPECIAL 3.25
PHASE LOCKED LOOPS		
XR-210	FSK Demodulator	5.20
XR-215	High Frequency PLL	6.60
XR-567CP	Tone Decoder (mini DIP)	1.95
XR-567CT	Tone Decoder (TO-5)	SPECIAL .99
STEREO DECODERS		
XR-1310P	PLL Stereo Decoder	3.20
XR-1310EP	PLL Stereo Decoder	3.20
XR-1800P	PLL Stereo Decoder	3.20
WAVEFORM GENERATORS		
XR-205	Waveform Generator	8.40
XR-2206CP	Monolithic Function Generator	SPECIAL 4.49
XR-2207CP	Voltage-Controlled Oscillator	3.85
OTHER EXAR IC'S		
XR-1468CN	Dual + 15V Tracking Regulator	SPECIAL 2.95
XR-1488N	Quad Line Driver	5.80
XR-1489AN	Quad Line Receiver	4.80
XR-2208CP	Operational Multiplier	5.20
XR-2211 CP	FSK Demodulator/Tone Decoder	6.70
XR-2261	Monolithic Proportional Servo IC System w/4 ea. Driver Transistor	3.79

Special Requested Items

RC4194	Dual Track V Reg	$5.95	N8T97	$3.00	2533	$7.95
RC4195	±15V Track Reg	3.25	4024P	2.25	8263	5.95
F9368	Decoder	3.95	2513	11.00	8267	2.75
LD110/111	DVM Chip Set	28.00	2518	7.00	8288	1.15
CA3130	Super CMOS Op Amp	1.49	2524	3.50	8826	3.00
MC1408L7	A/D	9.95	2525	6.00	8880	1.35
F3341	FIFO	8.95	2527	5.00	7497	5.00

ZENERS—DIODES—RECTIFIERS

TYPE	VOLTS	W	PRICE	TYPE	VOLTS	W	PRICE
IN746	3.3	400m	4/1.00	IN4003	200 PIV	1 AMP	.10
IN751A	5.1	400m	4/1.00	IN4004	400 PIV	1 AMP	.10
IN752	5.6	400m	4/1.00	IN3600	50	200m	6/1.00
IN753	6.2	400m	4/1.00	IN4148	75	10m	15/1.00
IN754	6.8	400m	4/1.00	IN4154	35	10m	12/1.00
IN965B	15	400m	4/1.00	IN4734	5.6	1w	.28
IN5232	5.6	500m	.28	IN4735	6.2	1w	.28
IN5234	6.2	500m	.28	IN4736	6.8	1w	.28
IN5235	6.8	500m	.28	IN4738	8.2	1w	.28
IN5236	7.5	500m	.28	IN4742	12	1w	.28
IN456	25	40m	6/1.00	IN4744	15	1w	.28
IN458	150	7m	6/1.00	IN1183	50 PIV	35 AMP	1.60
IN485A	180	10m	5/1.00	IN1184	100 PIV	35 AMP	1.70
IN4001	50 PIV	1 AMP	.09	IN1186	200 PIV	35 AMP	1.80
IN4002	100 PIV	1 AMP	.10	IN1188	400 PIV	35 AMP	3.00

TRANSISTORS

MPS-A05	5/1.00			PN4249	4/$1.00
MPS-A06	5/$1.00	PN3567	3/$1.00	PN4250	4/$1.00
2N2219A	3/$1.00	PN3568	4/$1.00	2N4400	4/$1.00
2N2221	4/$1.00	PN3569	4/$1.00	2N4401	4/$1.00
2N2222A	5/$1.00	2N3704	5/$1.00	2N4402	4/$1.00
2N2369	5/$1.00	2N3705	5/$1.00	2N4403	4/$1.00
2N2369A	4/$1.00	2N3700	5/$1.00	2N4409	5/$1.00
FN2415	5/$1.00	2N3707	5/$1.00	2N5086	4/$1.00
2N2484	4/$1.00	2N3711	5/$1.00	2N5087	4/$1.00
2N2906A	4/$1.00	2N3724	$.65	2N5088	4/$1.00
2N2907A	5/$1.00	2N3725	$1.00	2N5089	4/$1.00
2N2925	5/$1.00	2N3903	5/$1.00	2N5129	5/$1.00
2N3053	2/$1.00	2N3904	4/$1.00	2N5138	5/$1.00
2N3055	$.89	2N3905	4/$1.00	2N5139	5/$1.00
MJE3055	$.89	2N3906	4/$1.00	2N5209	5/$1.00
2N3392	5/$1.00	2N4013	3/$1.00	2N5951	5/$1.00
2N3398	5/$1.00	2N4014	3/$1.00	C10681 SCR	2/$1.00

CAPACITOR CORNER

50 VOLT CERAMIC DISC CAPACITORS

	1-9	10-49	50-100		1-9	10-49	50-100
10pf	.05	.04	.03	.001	.05	.04	.035
22 pf	.05	.04	.03	.0047	.05	.04	.035
47 pf	.05	.04	.03	.01	.05	.04	.035
100 pf	.05	.04	.03	.022	.06	.05	.04
220 pf	.05	.04	.03	.047	.06	.05	.04
470 pf	.05	.04	.035	.1	.12	.09	.075

100 VOLT MYLAR FILM CAPACITORS

.001mf	.12	.10	.07	.022mf	.13	.11	.08
.0022	.12	.10	.07	.047mf	.21	.17	.13
.0047mf	.12	.10	.07	.1mf	.27	.23	.17
.01mf	.12	.10	.07	.22mf	.33	.27	.22

± 20% DIPPED TANTALUMS (SOLID) CAPACITORS

.1	35V	.28	.23	.17	1.5	35V	.30	.26	.21
.15	35V	.28	.23	.17	2.2	25V	.31	.27	.22
.22	35V	.28	.23	.17	3.3	25V	.31	.27	.22
.33	35V	.28	.23	.17	4.7	25V	.32	.28	.23
.47	35V	.28	.23	.17	6.8	25V	.36	.31	.25
.68	35V	.28	.23	.17	10	25V	.40	.35	.29
1.0	35V	.28	.23	.17	15	25V	.63	.50	.40

MINIATURE ALUMINUM ELECTROLYTIC CAPACITORS

Axial Lead					Radial Lead				
.47	50	.15	.13	.10	.47	25	.15	.13	.10
1	50	.16	.14	.11	.47	50	.16	.14	.11
3.3	50	.15	.13	.10	1	16	.15	.13	.10
4.7	25	.16	.14	.12	1	25	.16	.14	.11
10	25	.15	.13	.10	1	50	.16	.14	.11
10	50	.16	.14	.12	4.7	16	.15	.13	.10
22	25	.17	.15	.12	4.7	25	.15	.13	.10
22	50	.24	.20	.18	4.7	50	.16	.14	.11
47	25	.19	.17	.15	10	16	.14	.12	.09
47	50	.25	.21	.19	10	25	.15	.13	.10
100	25	.24	.20	.18	10	50	.16	.14	.12
100	50	.35	.30	.28	47	50	.24	.21	.19
220	25	.32	.28	.25	100	16	.19	.15	.14
220	50	.45	.41	.38	100	25	.24	.20	.18
470	25	.33	.29	.27	100	50	.35	.30	.28
1000	16	.55	.50	.45	220	16	.23	.17	.16
2200	16	.70	.62	.55	470	25	.31	.28	.26

COMPUTER–DATA INPUT KEYBOARDS



B5283



B5199

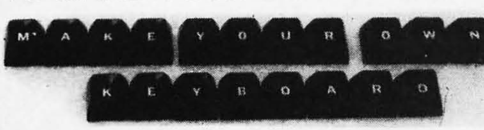
ASCII encoded keyboard. In its own enclosure. Originaly used in SANDERS ASSOCIATES 720 Terminal System. In like new condition. Usefull for any project requiring an ASCii encoded keyboard. 50 Alpha Numeric keys plus 11 computer symbols

STOCK NO.B5283 keyboard $35.00 2/65.00

MICRO–SWITCH (Honeywell) 8 bit binary coded board. 56 keys, alpha - meric and computer symbols Built in TTL decoder. New in factory cartons. A beautiful keyboard.

STOCK NO.B5199 Microswitch keyboard. $45.00 2/80.00

KEYTOPS & SWITCHES TO MAKE YOUR OWN KEYBOARD



We have a large selection of KEYTOPS and SWITCHES, made by RAYTHEON CO. The keytops come in black, grey and white, with contrasting legends. The switches mate with the tops, and are magnetic reed switches. The following combinations are available:

54 key typewriter set, keys only,	black	K9276	2.95
54 key typewriter set, keys only,	grey	K9278	2.95
54 key TTY set, no symbols	white	K9279	2.95
54 key TTY set, with symbols	white	K9282	2.95
54 key set, keys & switches	black	K9288	30.00
54 key set, keys & switches	grey	K9290	30.00
54 TTY set, no symbols keys & Sw.	White	K9291	30.00
54 TTY set, with symbols, keys & Sw.	white	K9291	30.00
11 Key Numeric set. Keys only	Black	K9283	1.50
11 Key Numeric set, Keys only	Grey	K9284	1.50
11 Key Numeric set, Keys only	White	K9295	1.50
12 Key numeric set, Keys only	white	K9286	1.50
11 Key Num.set, keys & switches	Black	K9293	7.00
11 Key Num. set, keys & switches	Grey	K9294	7.00
11 Key Num. set, keys & switches	White	K9295	7.00
12 Key Num. set, keys & switches	white	K9296	7.50
Blank key 1½ keys wide	white	K9297A	3/.25
Blank key 2 keys wide	white	K9297B	3/.25
K9297A with switch	white	K9298A	3/2.00
K9297B with switch	white	K9298B	3/2.00

MINIATURE 7 SEGMENT READOUT

Miniature 7 segment LED readout (EXITON XMN 101). Displays all numbers and 9 letters. O.D. 5/16"x 1/4" Display is .12 ". SPECIAL FOR THIS ISSUE ONLY

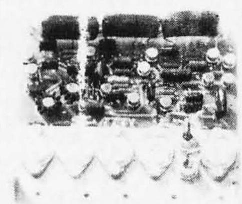
STOCK NO.B5173 with data sheet .50 ea. 5/2.00

TRANSFORMERS

Computer projects need power supplies. Finding the right power transformer can be a problem. We have one of the largest and most diversified stocks of power transformers in the country. Below we list some representative items in our inventory. Our catalog, free on request lists many more.

36 V. @ 1.0 A. ct, & 6.3 V @ 200 ma. 3 lbs.	B9313	3.50 2/6.00
70 V. @ 1.5 A. ct, & 6.3 V @ 500 ma. 6 lb.	B9314	6.50 2/12.00
90 V. @ 2.0 A. ct, & 6.3 V @ 1.5 A. 8½ lb.	B9315	9.95 2/19.00
50 V.=@ 1.5 A. ct, & 6.3V @500 ma. 6 lb.	B9316	6.50 2/12.00
26 V. @ 1.0 A. ct. &6.3 V. @ 500 ma. 3 lb.	B9318	3.75 2/7.00
38 V. @ 1.5 A. ct. & 6.0 V.@ 500 ma. 2 lb.	B9319	6.95 2/13.00
350 V. @ 35 ma. ct. & 6.3 V. @ 2.7 A 2 lb.	B9321	3.50 2/6.00
70 V. @ 1.5 A. ct. & 6.3 V @ 1.5 A. 7Lb.	B9322	6.75 2/13.00
35 V. @ 6.0 Ct. & 10 V. @ 10.0 A. 6.0 Lb.	B9906	8.95 ea.
64 or 32 V. @ 8.0 A. ct. & 18 V. @ 8.0 A ct. 10 lb.	B9905	11.95

VOLTAGE REGULATOR BOARDS





B5169 is a board containing 3 15 volt high current regulators with 0.1% regulation. 2 of the regulators are rated @ 3 Amps., and the other @ 6.0 amps. The current in each regulator may be doubled with the regulation going to 0.5%.All 3 regulators are short circuit proof, and 2 have electronic crowbar protection. Brand new, in factory boxes.

STOCK NO.B5169 $11.95 ea. 2/21.00

B9013 is a triple regulator with + 12 volt regulation @ 200 ma. and the third regulator is a tracking regulator, providing regulation from 0 to 5 volta @ .5 A.

STOCK NO.B9013 $5.95 ea. 2/10.00

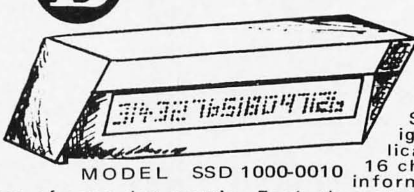
Both regulators above come withcircuit diagrams.

OPERATIONAL AMPLIFIERS
(OP —AMPS)

TYPE	DESCRIPTION	CASE	STOCK	PRICE
709	Hi Performance	TO-5	B4301	.50 5/2.00
4709	Dual 709	DIP	B5301	1.00 6/5.00
741	Hi Performance	DIP	B4316	.65 5/3.00
747	Dual 741	DIP	B4317	1.25 5/5.00
741	Hi Performance	Mini DIP	B4345	.65 5/3.00
747CT	Dual 741	TO-5	B3111	1.25 5/5.00
1458	Dual 741	Mini DIP	B3112	1.25 5/5.00
LM101A	Gen, Purpose	TO-5	B4503	.50 5/2.00

SELF SCAN PANEL DISPLAY

Burroughs



MODEL SSD 1000-0010

BURROUGHS SELF SCAN display, designed for numeric application, requiring up to 16 characters of numeric information. Display is made up of neon dot matrix. Each character is defined by a positive logic 4 bit code. Display operates in a scanning mode, scanning from left to right, one column at a time. Electronics is in interior of bezel, and consists of LSI chip and integrated circuits. Current distributor price is $135.00 . LIMITED QUANTITY

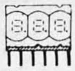
STOCK NO.5180 with data $49.50 2/90.00

Please include sufficient postage. Excess refunded MINIMUM ORDER $5.00

DELTA ELECTRONICS CO.

BOX 1, LYNN, MASSACHUSETTS 01903
Phone (617) 388-4705

Send for the latest edition of our catalog. Loaded with electronic and computer bargains.



7-Segment Readout 12-PIN DIP

Three digits with right-hand decimal
Plugs into DIP sockets
Similar to (LITRONIX) DL337
Magnified digit approximately .1"
Cathode for each digit
Segments are parallel for multiple operation
5-10 MA per segment
EACH $1.75   4 (12 DIGITS) $6.00)

RCA Numitron



EACH............$ 5.00

SPECIAL: 5 FOR $20.00

DR2010

MOS MEMORY 2102-2

1024 Bit Fully Decoded Static MOS Random Access Memory

- fast access 650ns
- fully TTL compatible
- n channel silicon gate
- single 5 volt supply
- tri-state output
- 1024 by 1 bit
- chip enable input
- no clocks or refreshing required

Brand New Factory Parts
16 PIN DIP Each $5.00
8 for $34.95

Power Supply SPECIAL!

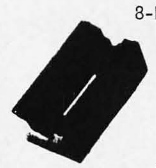
723 DIP variable regulator chip 1-40V, + or - output @ 150 MA 10A with external pass transistor--with diagrams for many applications.
EACH $1.00   10 FOR $8.95

5001 Calculator

40-Pin calculator chip will add, subtract, multiply, and divide. 12-digit display and calculate. Chain calculations. True credit balance sign output. Automatic over-flow indication. Fixed decimal point at 1, 2, 3, or 4. Leading zero suppression. Complete data supplied with chip.
CHIP AND DATA........ONLY $2.49
DATA ONLY (Refundable)... $1.00
5002 LOW POWER CHIP AND DATA $12.95

High Quality PCB Mounting IC Sockets

8-PIN, 14-Pin, 16-Pin and 24-Pin PCB mounting ONLY--no wire wrap sockets.



8-Pin.............$ .22
14-Pin........$ .26
16-Pin........$ .30
24-Pin........$ .75
40-Pin..........$1.25

All IC's are new and fully tested. Leads are plated with gold or solder. Orders for $5.00 or more will be shipped prepaid Add $ .35 for handling and postage for smaller orders; residents of California add sales tax. IC orders are shipped within 2 workdays--kits are shipped within 10 days of receipt of order. $10.00 minimum on C.O.D.'s.

Mail Orders to:
P.O. Box 41727
Sacramento, CA 95841
Phone (916) 334-2161

BABYLON ELECTRONICS

Money back guarantee on all goods!

Dale Trimmer

- 12 turn trimpots which plug into a DIP socket
- 5K and 200K
- ¼" x ¼" x ¼"
- 4 leads spaced .3" x .2"

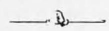
Each $1.00 10 for $8.95

1000 MHz Counter

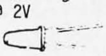
11C05 Fairchild 1GHz Divide By Four
- DC to 1000 MHz operation
- AC or DC coupled
- Voltage compensated
- TTL or ECL power supply
- 50 ohm drive output
- Lead compatible with Plessy SP613
- True and complement ECL outputs
- 14 pin DIP
- Data and application notes

Each $49.95

LED's



MV50 Red Emitting $ .20
10-4 MA @ 2V   10 FOR $1.25



MV5024 Red TO-18 $ .35
High Dome   10 FOR $2.95



MV10B Visible Red $ .30
5-7 MA @ 2V   10 FOR $2.50

CMOS

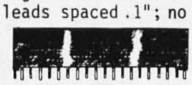
CD4001	$ .45	CD4023	$ .45
CD4002	.45	74C20	.65
CD4011	.45	74C160	3.25
CD4012	.45		

3-Amp Power Silicon Rectifiers

MARKED EPOXY AXIAL PACKAGE

PRV	PRICE	PRV	PRICE
100	$.10	800	$.30
200	.15	1000	.40
400	.18	1200	.50
600	.23	1500	.65

DIODE ARRAY 10-1N914 silicon signal diodes in one package. 20 leads spaced .1"; no common connections.

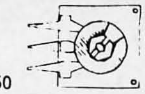


EACH.....$.29
10 FOR $2.50

7400	.20	74H51	.25
74H00	.30	7453	.20
7401	.20	7454	.20
74H01	.25	74L54	.25
7402	.25	74L55	.25
7403	.25	7460	.16
7404	.25	74L71	.25
74H04	.30	7472	.40
7405	.30	74L72	.60
7406	.40	7473	.35
7408	.30	74L73	.75
74H08	.30	7474	.45
7410	.20	74H74	.75
7413	.75	7475	.80
7417	.40	7476	.55
7420	.20	74L78	.70
74L20	.30	7480	.50
74H20	.30	7483	.70
74H22	.30	7489	3.00
7430	.20	7490	1.00
74H30	.30	7492	.65
74L30	.30	7493	1.00
7440	.20	7495	.65
74H40	.30	74L95	1.00
7442	1.00	74107	.35
7447	1.50	74145	1.25
7450	.20	74180	1.00
74H50	.30	74193	1.50
7451	.20	74195	.65

7400 Series DIP

25K Trimmer



PRINTED CIRCUIT BOARD TYPE
EACH $.20   10 FOR $1.50

Rectifiers



VARO FULL-WAVE BRIDGE

VS647	2A	600V	$1.10
MR810 Rectifier	50V	1A	$ .10

Special 811: Hex Inverter

TTL DIP Hex Inverter; pin interchangeable with SN 7404. Parts are brand new and branded Signetics and marked "811."

DATA SHEET SUPPLIED

EACH	$ .16
10 FOR	1.50
100 FOR	14.00
1000 FOR	110.00



1 AMP RECTIFIER

1N4007 1KV PRV
EACH $ .15
SALE 10 for $1.00



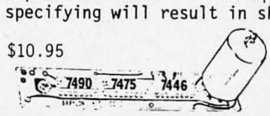
MAN 4



7-Segment, 0-9 plus letters. Right-hand decimal point. Snaps in 14-pin DIP socket or Molex. IC voltage requirements. Ideal for desk or pocket calculators!
EACH $1.20   10 OR MORE $1.00 EACH

CD-2 Counter Kit

This kit provides a highly sophisticated display section module for clocks, counters, or other numerical display needs. The unit is .8" wide and 4 3/8" long. A single 5-volt power source powers both the ICs and the display tube. It can attain typical count rates of up to 30 MHz and also has a lamp test, causing all 7 segments to light. Kit includes a 2-sided (with plated thru holes) fiberglass printed circuit board, a 7490, a 7475, a 7447, a DR2010 RCA Numitron display tube, complete instructions, and enough MOLEX pins for the ICs... NOTE: boards can be supplied in a single panel of up to 10 digits (with all interconnects); therefore, when ordering, please specify whether you want them in single panels or in one multiple digit board. Not specifying will result in shipping delay.
COMPLETE KIT ONLY $10.95
FULLY-ASSEMBLED UNIT $15.00



Boards supplied separately @ $2.50 per digit.

L I N E A R S

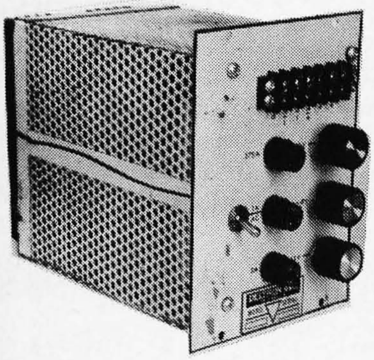
NE555	Precision timer	.90
NE560	Phase lock loop DIP	2.95
NE561	Phase lock loop DIP	3.00
NE565	Phase lock loop	2.95
NE566	Function generator TO-5	3.50
NE567	Tone decoder TO-5	3.50
709	Popular Op Amp DIP	.40
710	Voltage comparator DIP	.60
711	Dual comparator DIP	.45
723	Precision voltage regulator DIP	1.00
741	Op amp TO-5/MINI DIP	.45
748	Op Amp TO-5	.80
CA3018	2 Isolated transistors and a Darlington-connected transistor pair	1.00
CA3045	5 NPN transistor array	1.00
LM100	Positive DC regulator TO-5	1.00
LM105	Voltage regulator	1.25
LM302	Op Amp voltage follower TO-5	1.25
LM308	Op Amp TO-5	2.00
LM309H	5V 200 MA power supply TO-5	1.00
LM309K	5V 1A power supply module TO-3	1.00
LM311	Comparator Mini	1.75
LM370	AGC amplifier	1.75
LM380	2-Watt Audio Amp	1.75
LM1595	4-Quadrant multiplier	1.70
MC1536T	Op Amp	1.35

KEYBOARD WITH ENCODER

Late model keyboard used on computer terminal. Mounted in modern designed wood grained case dust enclosure for desk top use. Utilizes magnetic reed relay bounceless keyswitches. The encoder board mounted within. Fine biz for Morse Code Generator, TV typewriter, computer terminals, etc.

Ship wgt. 7 lbs. #SP-153-L $35.00





GENERAL PURPOSE POWER SUPPLY

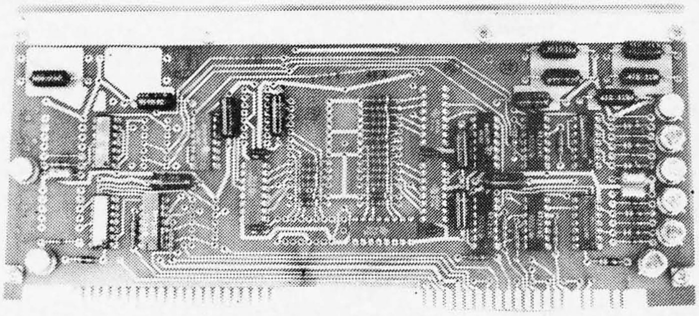
A well designed transistorized regulated power supply with many uses. Each voltage adjustable by a pot. Each voltage fused. 115 volts AC 60 cycle input. Output (minus) 12 volts at 3 amp, 12 volts (plus) at 1/3 amp, 6 volts at 1 amp . . . three output voltages. Many uses . . . as battery charger, op amp, 5 volt logic supply, operate your car radio or tape player, CB set, in the house, etc. A commercially built regulated supply for far less than the price of a kit.

Ship wgt. 10 lb #SP-152-L . . . . $12.50, 5/$50.00

MEMORY SYSTEM $125.00

New memory system by Honeywell, small . . . measures only 9x4x1 inches. 1024 core memory, 1024 words with 8,9,10 bits/word. Random access, with all logic, register, timing, control, core select and sense functions in one package. New, booklet of schematics and data. Looks like a good beginning for a mini-computer. Limited supply on hand.

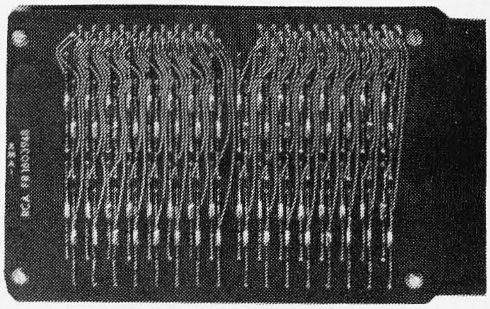
Ship wgt 3 lbs. #SP-79 . . . . $125.00



CORE MEMORY

This one by RCA, sandwich board construction with diode matrix. Fairly small measures approx. 5 x 7 inches. The memory is on one plane 32 x 32 x 9 (9216). Edge connector simplifies use. They are brand new and we have some data provided by one of our customers which we will photo copy and include.

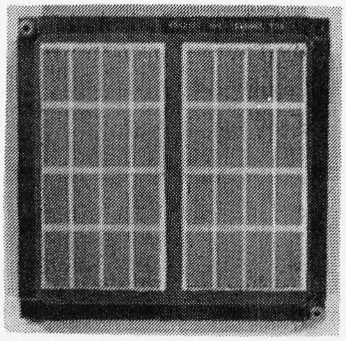
#SP-80 . . . . $25.00



CORE MEMORY

Another brand new memory, ultra small. Measures only 4 x 4 inches with format on one plane of 32 x 32 x 16 (16,384). Only about 35 units of this on hand.

#SP-81 . . . . $20.00



FREE CATALOG

Please add shipping cost on above.

Meshna

MESHNA PO Bx 62 E. Lynn Mass. 01904

BYTE reader service

To get further information on the products advertised in this issue of BYTE merely tear, rip, or snip out this advertiser index, fill out the data at the bottom of the page, mark the appropriate boxes, and send the works to BYTE, Peterborough NH 03458. Readers get extra Brownie Points for sending for information since this encourages advertisers to keep using BYTE – which in turn brings you a bigger BYTE.

ADVERTISER INDEX

Messages for the editor:

Reader's Service
BYTE
Green Publishing Inc.
Peterborough NH 03458

OCTOBER 1975

BYTE acquired via
☐ Subscription
☐ Newsstand
☐ Stolen

Please print or type.

Name ____________________

Address ____________________

City ____________ State ______ Zip ______

Coupon expires in 60 days . . .

THE BYTE QUESTIONNAIRE

BYTE is dedicated to the needs of its readership. In order to better gauge matters of editorial policy and content, as well as to give our advertisers some "hard facts", we publish this questionnaire. In this month's list are a few questions of editorial interest:

What is your primary interest in personal computing?

Would you suggest a game or two you'd like to see described in BYTE?

How did you hear about BYTE?

What did you like best about the first two BYTEs?

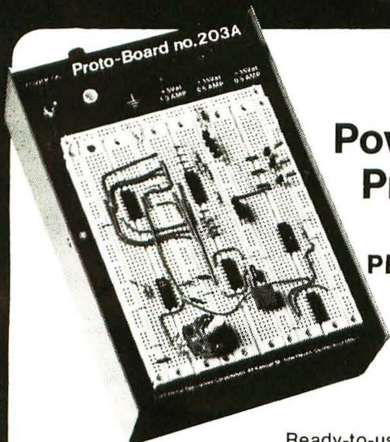
What did you like least about the first two BYTEs?

Are you a member of a computer club or society?

These questions are a short form "letter to the editor." If you have additional comments, don't hesitate to write! Send completed questionnaires to BYTE, Dept. Q, Peterborough NH 03458.

five new breadboard testers from CSC

Continental Specialties Corp. offers a total line of breadboard test devices . . . everything from inexpensive kits to high-power professional units and logic monitors too. Each high quality, compact unit comes with a guarantee of complete satisfaction or your money back within 10 days. Here are but five of the "hottest" items we make. . .



Power for the Professional!

New Proto Boards PB-203 and PB-203A with built-in regulated short-proof power supplies!

Ready-to-use. Just plug in and start building! 2 extra floating 5-way binding posts for external signals (PB-203 only). Completely self-contained with power switch, indicator lamp and power fuse. 24 14-pin DIP capacity. All metal construction . . . no chipping or cracking as with plastic cases. Two-tone quality case makes both PB-203 and PB-203A aesthetically, as well as technically attractive.

PB-203

- 3 QT-59S Sockets
- 4 QT-59B Bus Strips
- 1 QT-47B Bus Strip
- Fuse • Power Switch
- Power-On Light
- 9.75"L x 6.6"W x 3.25"H
- Weight: 5 lbs.
- 5V, 1 AMP regulated power supply

75.

Add $2.50 shipping/handling

OUTPUT SPECIFICATIONS

Output Voltage	5V ± ¼V
Ripple & Noise	@ ½ AMP 10 millivolts
Load Regulation	Better than 1%

PB-203A

- 3 QT-59S Sockets
- 4 QT-59B Bus Strips
- 1 QT-47B Bus Strip
- Fuse • Power Switch
- Power-On Light
- 9.75"L x 6.6"W x 3.25"H
- Weight: 5 lbs.
- 5V, 1 AMP regulated power supply (same as PB-203)
- +15V, ½ AMP regulated power supply
- −15V, ½ AMP regulated power supply

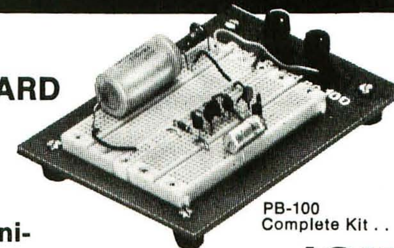
120.

Add $2.50 shipping/handling

OUTPUT SPECIFICATIONS

Output Voltage	15V, internally adjustable
Ripple & Noise	@ ¼ AMP, 10 millivolts
Load Regulation	Better than 1%

PROTO BOARD 100



A complete mini-breadboard budget kit with full IC capacity

PB-100 Complete Kit . . .

19.95

Add $1.50 shipping/handling

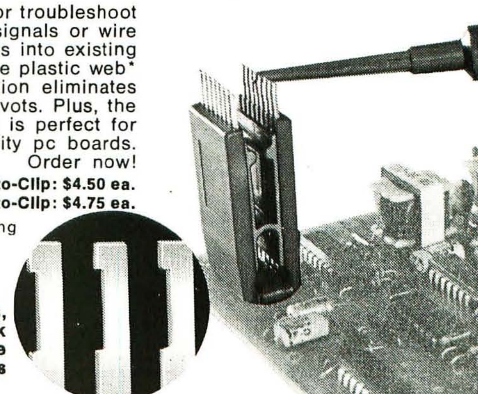
The PB-100 is a low cost, big 10 IC capacity breadboard kit, complete down to the last nut, bolt and screw. Includes 2 QT-35S Sockets; 1 QT-35B Bus Strip; 2 5-way binding posts; 4 rubber feet; screws and easy assembly instructions. 4.50" (114.3mm) wide x 6.00" (152.4mm) long x 1.35" (34.3mm) high. Order your PB-100 kit! Start building and testing now!

CSC PROTO-CLIP offers power-on . . . hands-off signal tracing . . . under $5!

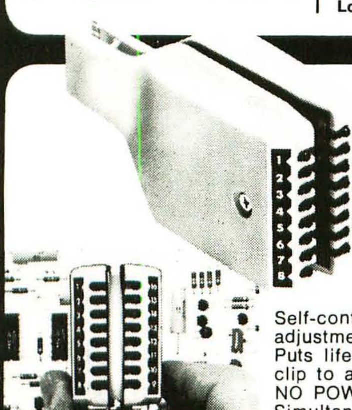
Trace signals or troubleshoot fast. Inject signals or wire unused circuits into existing boards. Flexible plastic web* construction eliminates springs and pivots. Plus, the narrow throat is perfect for high density pc boards. Order now!

PC-14 14-pin Proto-Clip: $4.50 ea.
PC-16 16-pin Proto-Clip: $4.75 ea.

Add $1.00 shipping and handling



Scope probes, test leads lock onto unique toothed grips



Continental Specialties Corp. LOGIC MONITOR brings ICs to life faster than a scope . . . safer than a voltmeter

LM-1 **84.95 each**

Add $2.50 shipping/handling

Self-contained, pocket size. No adjustments or calibrations needed. Puts life into digital designs. Just clip to any DIP IC up to 16 pins. NO POWER SUPPLY NEEDED! Simultaneously displays static and dynamic logic states of DTL, TTL, HTL or CMOS on 16 large high intensity LEDs. Watch signals work through counters, shift registers, timers, adders, flip flops, decoders, entire systems. Concentrate on signal flow and input/output truth tables. Forget probe grounds, pin counting or sync polarity. Precision plastic guides and flexible plastic web* insure positive connections. Versatile. Fast. Accurate. Indispensable. Order yours today!

All Continental Specialties breadboard test devices are made in the USA, and are available off-the-shelf from your local distributor or CSC. Direct purchases may be charged on BankAmericard, Master Charge or American Express. You get a FREE English/Metric conversion slide rule with each order. Foreign orders please add 10% for shipping/handling. Prices are subject to change. Write or phone for complete illustrated catalog, plus the name and address of the CSC dealer nearest you.

*Patents Pending

CSC

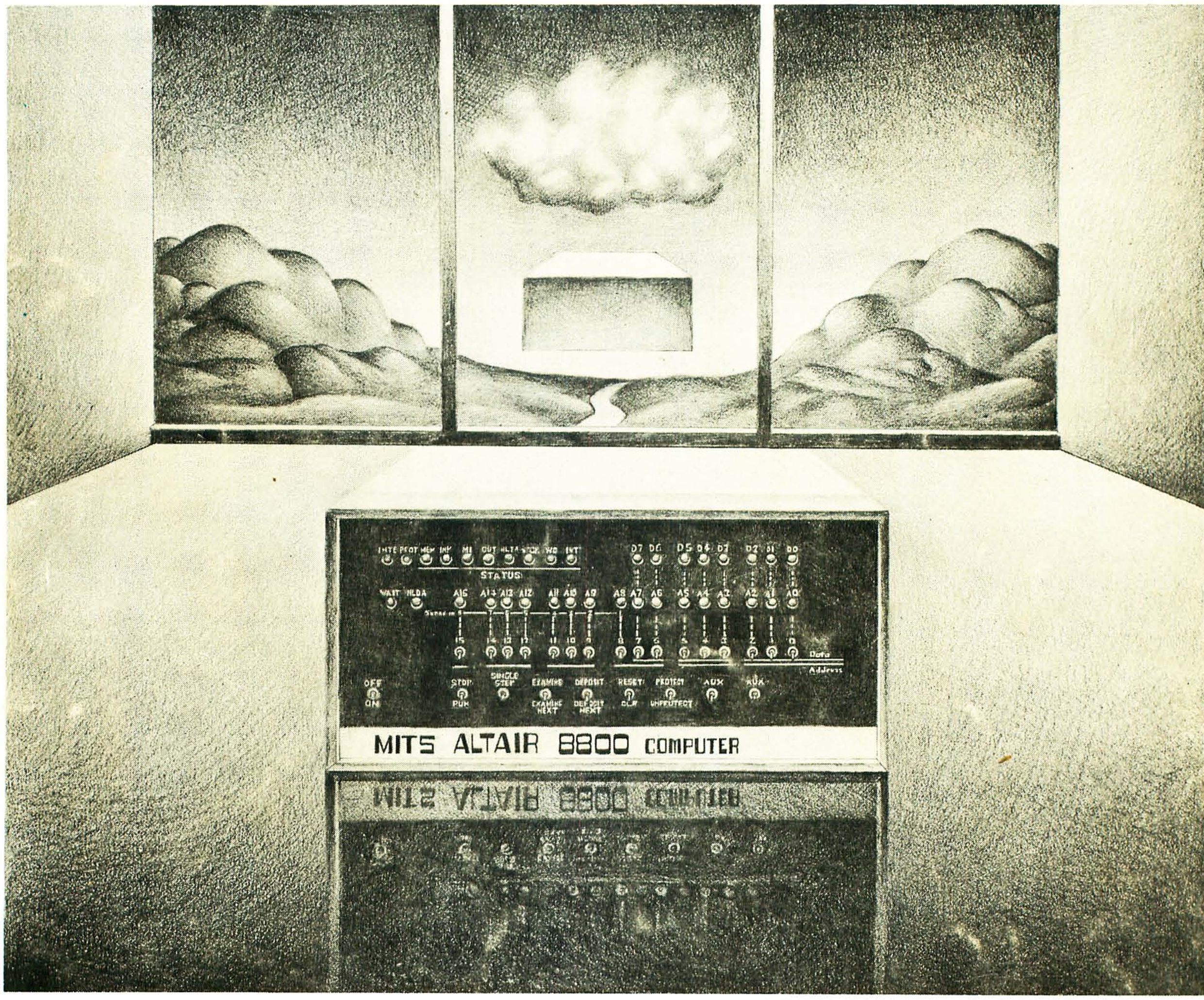
CONTINENTAL SPECIALTIES CORP.

44 Kendall St., Box 1942, New Haven, CT 06509 • 203/624-3103

West Coast Office: Box 7809, San Francisco, CA 94119 • 415/383-4207

CANADA: Available thru Len Finkler Ltd., Ontario

Created by Man.



The Affordable Computer.